Proceedings

THIRTY-THIRD ANNUAL ALLERTON CONFERENCE ON COMMUNICATION, CONTROL, AND COMPUTING

October 4 - October 6, 1995

Allerton House, Monticello, Illinois
Sponsored by the
Coordinated Science Laboratory and the
Department of Electrical and Computer Engineering of the
University of Illinois at Urbana-Champaign

PROCEEDINGS

THIRTY-THIRD ANNUAL ALLERTON CONFERENCE ON COMMUNICATION, CONTROL, AND COMPUTING

Sean Meyn
Ken Zeger
Conference Co-Chairs

Conference held
October 4 - October 6, 1995
Allerton House
Monticello, Illinois

Sponsored by
The Coordinated Science Laboratory
and
The Department of Electrical and Computer Engineering
of the
UNIVERSITY OF ILLINOIS
at
Urbana-Champaign

Digitized by the Internet Archive
in 2026 with funding from
University of Illinois Urbana-Champaign

https://archive.org/details/proceedings1995alle_0

621.3811
AL5
1995

This *Proceedings* contains papers presented at the Thirty-third Annual Allerton Conference on Communication, Control, and Computing which was held at Allerton House, Monticello, Illinois, on October 4, 5, and 6, 1995. As in past years, the Conference was sponsored by the Coordinated Science Laboratory and the Department of Electrical and Computer Engineering of the University of Illinois at Urbana-Champaign.

Eighty-five invited papers and forty-seven contributed papers were presented at the Conference in twenty-nine sessions. We thank the authors for contributing to the success of the Conference. We are also very grateful to our colleagues at the University of Illinois at Urbana-Champaign: Professors B. Bamieh, T. Başar, R. Blahut, B. Hajek, D. Jones, U. Madhow, J. Medanic, W. Perkins, D. Sarwate, R. Sreenivas, A. Vardy, P. Voulgaris and Drs. B. Hochwald, T. Linder, R. Sharma, and V. Tarokh for organizing many of the invited sessions, reviewing the contributed papers, chairing sessions, and helping in many other ways in the smooth running of the Conference. Invited sessions were also organized by Professors A. Kuh of the University of Hawai'i, H. Ozbay of Ohio State University, R. Pickholtz of George Washington University, B. Rimoldi of Washington University, C. Schlegel of the University of Texas at San Antonio, W. Szpankowski of Purdue University, and Dr. E. Altman of INRIA, Sophia Antipolis, France, and we are especially grateful to them for their efforts on behalf of the Conference. Professor T. Fuja of the University of Maryland was able to fill in at the last minute and chair a contributed session and for this we offer many thanks.

The plenary speaker at the Conference was Dr. Alan Laub, Chair, Department of Electrical and Computer Engineering, University of California at Santa Barbara, who gave a lecture titled "The Matrix Sign Function." We are grateful to Dr. Laub for sharing his thoughts on this topic with us. His excellent presentation contributed much to the success of the Conference. We are also indebted to the "Squeeze Tones" for providing musical entertainment at the Conference.

We thank Mrs. Beverly Becker and her staff at Allerton House for cheerfully accommodating our Conference and for their constant attention to our needs. It is also a great pleasure to thank all the staff members of the Coordinated Science Laboratory who so ably handled all the practical aspects of organizing the Conference. In particular, we wish to mention Mr. Steven Williams and Ms. Susan Baxley who handled all the business affairs, and Mr. Dan Jordan who handled the sales and shipping of these *Proceedings.* Most of all, however, we want to thank our Conference Secretary, Ms. Sheree Waltz, who with the able assistance of Ms. Becky Lonberger and Ms. Francie Bridges worked all year round in handling announcements, calls for papers, program preparation, Allerton House arrangements, registration, and the final editing of these *Proceedings.* The lion's share of the credit for the success of the Conference belongs to these ladies.

Sean Meyn and Ken Zeger

TABLE OF CONTENTS

ACKNOWLEDGMENTS i

I-A: MULTIUSER COMMUNICATIONS I

Organizers: B. Rimoldi, Washington University, St. Louis
C. Schlegel, Univ. of Texas at San Antonio
Chairman: B. Rimoldi
Washington University, St. Louis

COMBINED ERROR CORRECTION AND MULTIUSER DETECTION FOR RAYLEIGH FADING SYNCHRONOUS CDMA CHANNELS 01
A. El-Ezabi and A. Duel-Hallen

MULTIUSER DETECTION FOR FADING MULTIPATH MULTIUSER CHANNELS: A UNIFIED APPROACH TO DESIGN AND SUCCINCT EIGENBOUNDS ON PERFORMANCE: PART I 11
M.K. Varanasi

A NEW CODING METHOD FOR THE ASYNCHRONOUS MULTIPLE ACCESS CHANNEL 21
A.J. Grant and L.K. Rasmussen

A LINEAR RECEIVER FOR THE CODED ASYNCHRONOUS MULTIUSER CDMA CHANNEL 29
P.D. Alexander, L.K. Rasmussen, and C. Schlegel

DECODING DELAY OF RATE-SPLITTING MULTIPLE ACCESSING WITH BURSTY SOURCES 39
S.P. Fedortsov and B. Rimoldi

I-B: CONTROL SYSTEMS I

Chairman: B. Bamieh
University of Illinois at Urbana-Champaign

IMPROVING THE ROBUSTNESS OF LINEAR PARAMETER VARYING SYSTEMS USING PARAMETER FEEDBACK 49
D. Malloy and B.C. Chang

ROBUST STABILITY OF LINEAR SYSTEMS DEPENDING ON UNCERTAIN, SLOWLY-VARYING PARAMETERS 59
F. Amato, M. Corless, M. Mattei, and R. Setola

A NEW STRATEGY FOR RECONFIGURABLE CONTROL SYSTEMS 69
I.K. Konstantopoulos and P.J. Antsaklis

ON THE H_∞ CONTROL OF BILINEAR SYSTEMS USING TIME DOMAIN APPROACH 79
S.-P. Shue, M.E. Sawan, and K. Rokhsaz

ROBUST POLYHEDRAL STABILITY VIA EXTREME POINTS RESULTS 89
F. Amato, A. Caruso, F. Garofalo, and L. Glielmo

I-C: DISCRETE EVENT DYNAMIC SYSTEMS I

Organizer and Chairman: R. Sreenivas
University of Illinois at Urbana-Champaign

SUPERVISORY CONTROL SYNTHESIS FROM METRIC TEMPORAL LOGIC SPECIFICATIONS ... 96
M. Barbeau, F. Kabanza, and R. St-Denis

STRONG CONSISTENCY OF INFINITESIMAL PERTURBATION ANALYSIS FOR SYMMETRIC ATM INTREE NETWORKS ... 106
C.A. Brooks and P. Varaiya

APPLICATIONS OF ORDINAL OPTIMIZATION TO A ROBOT MOTION PLANNING PROBLEM ... 116
C.-H. Chen and V. Kumar

FAULT DETECTION AND IDENTIFICATION IN COMMUNICATION NETWORKS: A DISCRETE EVENT SYSTEMS APPROACH ... 126
Y. Park and E.K.P. Chong

ON OBSERVABILITY AND NONDETERMINISM IN DISCRETE EVENT CONTROL ... 136
M. Heymann and F. Lin

II-A: DISCRETE EVENT DYNAMIC SYSTEMS II

Organizer and Chairman: R. Sreenivas
University of Illinois at Urbana-Champaign

FORBIDDEN STATE AVOIDANCE IN CONTROLLED PETRI NETS UNDER PARTIAL OBSERVATION ... 146
L. Zhang and L.E. Holloway

A DISCRETE-EVENT SYSTEMS APPROACH TO MODELING DEXTROUS MANIPULATION ... 156
S.L. Ricker, N. Sarkar, and K. Rudie

ASSIGNMENT PROGRAM MODEL FOR CONTROL OF INFINITE STATE SYSTEMS ... 166
R. Kumar and V.K. Garg

PETRI NET SUPERVISORS FOR DES IN THE PRESENCE OF UNCONTROLLABLE AND UNOBSERVABLE TRANSITIONS ... 176
J.O. Moody and P.J. Antsaklis

II-B: MULTIOBJECTIVE CONTROL

Organizer and Chairman: P. Voulgaris
University of Illinois at Urbana-Champaign

MULTIOBJECTIVE CONTROL VIA INFINITE-DIMENSIONAL LMI OPTIMIZATION ... 186
N. Elia, P.M. Young, and M.A. Dahleh

ON REDUCED ORDER MIXED-NORM CONTROL SYNTHESIS: AN AIRCRAFT TERRAIN FOLLOWING EXAMPLE ... 196
D.R. Jacques and D.B. Ridgely

SOLUTION OF MIXED $H_2/H_\infty/l_\infty$ CONTROL PROBLEMS USING FFTs AND ALTERNATING PROJECTIONS ... 206
K.M. Grigoriadis

II-C: COMMUNICATION SYSTEMS

Chairman: D. Sarwate
University of Illinois at Urbana-Champaign

PARALLEL INTERFERENCE CANCELLATION FOR CDMA SYSTEMS ... 216
V. Ghazi-Moghadam, L.B. Nelson, and M. Kaveh

TCM/SSMA COMMUNICATION SYSTEMS WITH CASCADED-SEQUENCES AND PAM/QAM SIGNAL SETS ... 225
T.-T. Chen and J.S. Lehnert

FRAME SYNCHRONIZATION FOR PILOT SYMBOL ASSISTED MODULATION ... 235
J.A. Gansman, M.P. Fitz, and J.V. Krogmeier

A FRACTIONALLY SPACED BLIND BAYESIAN CONDITIONAL DECISION FEEDBACK ESTIMATOR WITH SYMBOL TIMING CONSIDERATION ... 245
I. Kang, S.B. Gelfand, and M.P. Fitz

TEMPORAL MULTIPLEXING GAINS ... 255
I. Sidhu, H. Jiang, and S. Jordan

II-D: LOSSY SOURCE CODING

Chairman: T. Linder
University of Illinois at Urbana-Champaign

PROGRESSIVE IMAGE TRANSMISSION ON A CHANNEL WITH MEMORY ... 265
B.S. Srinivas, E.A. Riskin, R. Ladner, and M. Azizŏglu

ACCOMMODATING ANATOMICAL VARIABILITY IN FUNCTIONAL IMAGING VIA DEFORMABLE TEMPLATES ... 275
A. Banerjee, G.E. Christensen, J. Haller, S. Joshi, M.E. Raichle, and M.I. Miller

TIME-FREQUENCY ANALYSIS USING DYNAMIC STATISTICAL MODELS WITH APPLICATIONS TO SPEECH ANALYSIS ... 285
S. Lu and P.C. Doerschuk

AN ITERATIVE APPROACH TO VECTOR QUANTIZATION WITH VERY SMALL CODEBOOKS .. 295
A. Narayan and J.F. Doherty

A LAYERED VIDEO CODING ALGORITHM FOR MULTIMEDIA APPLICATIONS OVER ATM...... 303
P. Gburzyński and G. Shankar

II-E. TOPICS IN CODING

Chairman: V. Tarokh
University of Illinois at Urbana-Champaign

ASYMPTOTIC CODING GAIN AND TRELLIS COMPLEXITY ... 313
W. Lin and R.J. McEliece

MSN CODES FOR TWO-TRACK, TWO-HEAD MAGNETIC RECORDING SYSTEMS ... 323
E. Kurtas, J.G. Proakis, and M. Salehi

AN APPROXIMATE CALCULATION METHOD OF THE PROBABILITY OF UNDETECTED ERROR FOR BINARY EXPANDED CONCATENATED CODES 333
T. Nishijima and S. Hirasawa

II-F: CONTROL SYSTEMS II

Chairman: T. Başar
University of Illinois at Urbana-Champaign

IDENTIFICATION OF LINEAR MULTIVARIABLE DYNAMICAL SYSTEMS WITH STATIC INPUT NONLINEARITIES 340
H. Al-Duwaish, M. Nazmul Karim, and V. Chandrasekar

SOME STABILITY RESULTS FOR NONLINEAR SAMPLED-DATA CONTROL SYSTEMS 349
L. Hou, A.N. Michel, and H. Ye

III-A: CODING THEORY I: TURBO CODES AND CONVOLUTIONAL CODES

Organizers: A. Vardy and R. Blahut
University of Illinois at Urbana-Champaign
Chairman: R. Blahut
University of Illinois at Urbana-Champaign

ON SELECTING CODE GENERATORS FOR TURBO CODES 357
J. Seghers, L.C. Perez, and D.J. Costello, Jr.

THE TURBO DECISION ALGORITHM 366
R.J. McEliece, E.R. Rodemich, and J.-F. Cheng

CONVOLUTIONAL CODES OVER GROUPS: A PRAGMATIC APPROACH 380
T. Mittelholzer

ASCETIC CONVOLUTIONAL CODES 382
Ø. Ytrehus

III-B: CONTROL SYSTEMS III

Chairman: W. Perkins
University of Illinois at Urbana-Champaign

OPTIMIZATION AND STABILITY ROBUSTNESS OF LARGE-SCALE LINEAR DISCRETE-TIME SYSTEMS 391
T.-c. Wang and M.E. Sawan

DESIGN OF OPTIMAL SAMPLED-DATA REPETITIVE CONTROL SYSTEMS 401
A. Langari and B.A. Francis

LINEAR DYNAMIC l^1-OPTIMAL CONTROLLERS FOR DISCRETE-TIME SISO SYSTEMS REVISITED 411
M.E. Alpay and M.H. Shor

III-C: FILTERING

Chairman: D. Jones
University of Illinois at Urbana-Champaign

UTILIZATION OF THE MV SPECTRUM FOR DIAGNOSIS AND ESTIMATION OF HARMONIC SIGNALS 421
A. Sexton and D.E. Lyon

OPTIMAL IMPULSE RESPONSE SHORTENING 431
P.J.W. Melsa, R.C. Younce, and C.E. Rohrs

III-D: ALGORITHMS AND COMPUTATIONS

Chairman: J. Medanic
University of Illinois at Urbana-Champaign

PARALLEL MATRIX SIGN ITERATIONS 440
B. Bakkaloğlu and Ç.K. Koç

PARAMETERIZED PARALLEL SUCCESSIVE OVER-RELAXATION ALGORITHMS FOR LINEAR ARRAYS 447
G.G.L. Meyer and M. Pascale

CHECKING REGULARITY OF INTERVAL MATRICES BY SIGNED DIGRAPHS 458
A. Nogi and Y. Ishida

A FIXED-SLOPE UNIVERSAL SEQUENTIAL ALGORITHM FOR LOSSY SOURCE CODING BASED ON GOLD-WASHING MECHANISM 466
Q. Liu, E.-h. Yang, and Z. Zhang

IV-A: CODING THEORY III: CODES OVER RINGS AND ALGEBRAIC GEOMETRY CODES

Organizers: A. Vardy and R. Blahut
University of Illinois at Urbana-Champaign
Chairman: R. Blahut
University of Illinois at Urbana-Champaign

A 2-ADIC APPROACH TO THE ANALYSIS OF CYCLIC CODES 475
A.R. Calderbank, W.-C. W. Li, and B. Poonen

ON GALOIS RINGS: A HASSE-DAVENPORT-TYPE RELATION FOR Z_4 KERDOCK CODE AND APPLICATIONS 480
T. Helleseth, P.V. Kumar, A.G. Shanbhag, and K. Yang

IMPROVED BEZOUT THEOREM AND ITS APPLICATIONS IN CODING THEORY 483
G.-L. Feng, T.R.N. Rao, and G.A. Berg

THE ALGEBRAIC STRUCTURE OF SOME AG GOPPA CODES 492
J. Little

IV-B: DISCRETE EVENT DYNAMIC SYSTEMS III

Organizer and Chairman: R. Sreenivas
University of Illinois at Urbana-Champaign

SUPERVISORY CONTROL OF FINITE AUTOMATA UNDER DYNAMIC PREDICATE SPECIFICATIONS 501
S.L. Chen and W.M. Wonham

THE DESIGN OF CORRIDOR POLICIES WITH SEPARABLE AVERAGE CRITERIA 510
C. Humes, Jr., L. de Oliveira Brandão, and M.P. Garcia

PLC IMPLEMENTATION OF A DES SUPERVISOR FOR A MANUFACTURING TESTBED 519
R.J. Leduc and W.M. Wonham

IV-C: LEARNING AND NEURAL NETWORKS

Chairman: B. Hochwald
University of Illinois at Urbana-Champaign

DENSITY ESTIMATION USING NEURAL NETWORKS 529
D.S. Modha and E. Masry

QUALITATIVE ANALYSIS OF COHEN-GROSSBERG NEURAL NETWORKS WITH MULTIPLE DELAYS 535
H. Ye, A.N. Michel, and K. Wang

COMBINATIONS OF WEAK CLASSIFIERS: A CASE ANALYSIS 545
C. Ji

IV-D: COMPUTER COMMUNICATION NETWORKS

Chairman: B. Hajek
University of Illinois at Urbana-Champaign

FANOUT SPLITTING IN NONBLOCKING COPY NETWORKS WITH SHARED BUFFERING 552
A. Chandra, H. Saidi, P.S. Min, and M.V. Hegde

ON ACHIEVING A THROUGHPUT OF ONE FOR A RANDOM ACCESS CHANNEL WITH COLLISIONS OF KNOWN MULTIPLICITY 562
S.S. Panwar and W. Grote

FORMAT INSENSITIVE PACKET SYNCHRONIZER 568
C.E. Love and H.F. Jordan

ROUTING SCHEMES FOR MULTIPLE RANDOM BROADCASTS IN ARBITRARY NETWORK TOPOLOGIES 578
E. Varvarigos and A. Banerjee

THROUGHPUT AND DELAY FOR COPY NETWORKS WITH INTERNAL BUFFERS 588
P.S. Min, M.V. Hegde, A. Chandra, and A. Maunder

IV-E: COMMUNICATION CHANNELS

Chairman: T. Fuja
University of Maryland

A CANONICAL FINITE-LENGTH MMSE-DFE 598
N. Al-Dhahir, J.M. Cioffi, and G.D. Forney, Jr.

LEAST-FAVORABLE DIVERGENCE-CONSTRAINED NOISE FOR BINARY-INPUT CHANNELS 608
A. McKellips and S. Verdú

OPTIMAL AND NEAR-OPTIMAL JOINT CHANNEL AND DATA ESTIMATION FOR FREQUENCY-SELECTIVE RAYLEIGH FADING CHANNELS 618
Y. Zhang, M.P. Fitz, and S.B. Gelfand

JOINT SIGNALING STRATEGIES FOR MAXIMIZING THE CAPACITY OF TWISTED PAIR LOOPS 628
A. Sendonaris, V. Veeravalli, and B. Aazhang

TIGHT UPPER BOUNDS ON THE BIT ERROR PROBABILITY FOR DIVERSITY RECEPTION IN SLOW RAYLEIGH FADING 638
S.A. Al-Semari and T.E. Fuja

ENUMERATION OF MARKOV CHAINS AND ERROR STATISTICS FOR GILBERT-ELLIOTT CHANNELS 648
C. Pimentel and I.F. Blake

IV-F: NON-RIGID MOTION TRACKING AND ROBOT CONTROL

Organizer and Chairman: R. Sharma
University of Illinois at Urbana-Champaign

NONRIGID MOTION TRACKING 658
C. Kambhamettu

A SYSTEMATIC APPROACH TO REASON ABOUT ACTIONS, PLANNING AND ROBOT CONTROL 668
C. Baral

VISUALLY SERVOED MANIPULATION USING AN ACTIVE CAMERA 678
B.J. Nelson and P.K. Khosla

V-A: CODING THEORY III: SOFT DECISION DECODING AND TRELLIS COMPLEXITY

Organizers: A. Vardy and R. Blahut
University of Illinois at Urbana-Champaign
Chairman: A. Vardy
University of Illinois at Urbana-Champaign

NEAR-MAXIMUM-LIKELIHOOD DECODING OF LINEAR CODES 688
I. Dumer

HEURISTIC SEARCH ALGORITHMS FOR SOFT-DECISION DECODING 690
C.-C. Shih, C.R. Wulff, C.R.P. Hartmann, and C.K. Mohan

A RECURSIVE MAXIMUM LIKELIHOOD DECODING PROCEDURE FOR A LINEAR BLOCK CODE USING AN OPTIMUM SECTIONALIZED TRELLIS DIAGRAM 700
T. Fujiwara, H. Yamamoto, T. Kasami, and S. Lin

ON THE LABEL GROUP COMPLEXITY OF LATTICES 710
V. Tarokh and I.F. Blake

ON CODES SATISFYING THE DOUBLE CHAIN CONDITION 714
T. Kløve

V-B: TELECOMMUNICATION NETWORKS: MODELING AND ANALYSIS I

Organizer and Chairman: E. Altman
INRIA, Sophia Antipolis, France

FAIR BLOCKING MECHANISMS FOR DISCRETE-TIME MULTIPLEXING 724
A.E. Conway and C. Rosenberg

ASYMPTOTIC ESTIMATES FOR BLOCKING PROBABILITIES IN A LARGE MULTI-RATE LOSS NETWORK 726
A. Simonian, J.W. Roberts, F. Théberge, and R. Mazumdar

ON THE CONTROL OF TANDEMS OF QUEUES WITH FINITE BUFFERS 736
G. Koole and Z. Liu

SUBEXPONENTIAL ASYMPTOTICS OF A NETWORK MULTIPLEXER 746
P.R. Jelenković and A.A. Lazar

ASYMPTOTIC BEHAVIOR OF A RELAXED SEQUENTIAL FLOW-CONTROL ALGORITHM FOR MULTICLASS NETWORKS 756
C.M. Rump and S. Stidham, Jr.

EFFECTIVE BANDWIDTH OF MARKOV FLUIDS WITH OCCUPANCY-BASED ADMISSION CONTROL 766
R.S. Maier

V-C: WIRELESS COMMUNICATIONS I

Organizer and Chairman: U. Madhow
University of Illinois at Urbana-Champaign

SPATIAL AND TEMPORAL PRECODING AND EQUALIZATION USING ADAPTIVE ARRAY ANTENNA FOR MOBILE RADIO COMMUNICATIONS 776
R. Kohno

THE MULTI-ACCESS FADING CHANNEL: SHANNON AND DELAY LIMITED CAPACITIES 786
S.V. Hanly and D.N. Tse

EVALUATION OF FIELD MEASUREMENTS FROM AN EXPERIMENTAL SLOW FREQUENCY HOP PERSONAL COMMUNICATION SYSTEM 796
D.M. Tolli, P.D. Rasky, G.M. Chiasson, and J.F. Kepler

LIMITS ON PERFORMANCE WITH RANDOM POLYPHASE SEQUENCES IN DS/SSMA SYSTEMS 805
T.M. Lok and J.S. Lehnert

THROUGHPUT OPTIMIZATION IN FADED MULTICARRIER SYSTEMS 815
D.L. Goeckel and W.E. Stark

DATA COMMUNICATIONS IN A CODE-DIVISION MULTIPLE-ACCESS SYSTEM 825
D.L. Noneaker and M.B. Pursley

V-D: ROBUST CONTROL OF NONLINEAR AND INFINITE DIMENTIONAL SYSTEMS

Organizer and Chairman: H. Özbay
Ohio State University

ON SOME LMI PROBLEMS ARISING IN NONLINEAR STOCHASTIC CONTROL 835
E.E. Yaz

ROBUSTNESS AND L^∞ PERFORMANCE OF NONLINEAR FEEDBACK SYSTEMS 845
I.J. Fialho and T.T. Georgiou

ON H^∞ OPTIMAL SAMPLED-DATA REPETITIVE CONTROLLERS 853
T.E. Peery and H. Özbay

MINIMUM OUTPUT VARIANCE CONTROL FOR FSN MODELS: CONTINUOUS-TIME CASE 863
R.E. Skelton and G. Shi

V-E: DATA COMPRESSION

Organizer and Chairman: W. Spankowski
Purdue University

TOWARD SOFTWARE-ONLY LOW-DELAY VIDEO ENCODING 873
T. Berger and Y.-J. Chiu

STRING MATCHING DATA COMPRESSION ALGORITHMS: PERFORMANCE ANALYSIS 881
E.-h. Yang and J.C. Kieffer

ON GENERALIZED DIGITAL SEARCH TREES WITH APPLICATIONS TO A GENERALIZED LEMPEL-ZIV ALGORITHM 891
W. Szpankowski and J. Tang

THE REDUNDANCY OF SOURCE CODING WITH A FIDELITY CRITERION 901
Z. Zhang, E.-h. Yang, and V.K. Wei

VI-A: CODING THEORY IV: STRUCTURE OF CODES

Organizers: A. Vardy and R. Blahut
University of Illinois at Urbana-Champaign
Chairman: A. Vardy
University of Illinois at Urbana-Champaign

THE ASSOCIATION SCHEME OF QUATERNARY PREPARATA CODES 908
A. Bonnecaze and I.M. Duursma

A CLASS OF FREQUENCY HOP CODES WITH NEARLY IDEAL CHARACTERISTICS FOR MULTIPLE-TARGET RECOGNITION 911
O. Moreno and S. Marić

VI-B: WIRELESS COMMUNICATIONS II

Organizer: U. Madhow
University of Illinois at Urbana-Champaign
Chairman: D. Noneaker
Clemson University

CAPACITY AND DYNAMIC RESOURCE ALLOCATION IN BROADCAST FADING CHANNELS 915
A. Goldsmith

RESOURCE ALLOCATION FOR PACKET DATA TRANSMISSION IN DS-CDMA 925
M.L. Honig and J.B. Kim

ALGORITHMS FOR POWER-EFFICIENT IDENTIFICATION OF RADIO-FREQUENCY IDENTIFICATION TAGS 935
A. Krishna and D.D. Kandlur

AN ALGORITHM FOR MINIMIZING QUEUING DELAY OF PACKETS IN CELLULAR DATA NETWORKS 945
J.M. Harris and S.P. Kumar

RESOURCE ALLOCATION FOR MOBILE USERS IN ATM NETWORKS 955
B.V. Patel and M. Schwartz

VI-C: TELECOMMUNICATION NETWORKS: MODELING AND ANALYSIS II

Organizer and Chairman: E. Altman
INRIA, Sophia Antipolis, France

THE LOSS PROCESS OF MESSAGES WITH REDUNDANT PACKETS IN AN M/M/1/K QUEUE 964
O.A. Hellal, E. Altman, and A. Jean-Marie

ACCESS CONTROL WITH DELAYED INFORMATION 974
E. Altman and D. Kofman

DESIGNING A MULTICAST SWITCH SCHEDULER 984
B. Prabhakar and N. McKeown

VI-D: NEURAL NETWORKS AND NONPARAMETRIC LEARNING

Organizer and Chairman: A. Kuh
University of Hawai'i

EM ALGORITHM AND RECURRENT NETWORKS 994
S. Ma and C. Ji

THE COMPLEXITY OF LEARNING FROM A MIXTURE OF LABELLED AND UNLABELLED EXAMPLES 1002
J. Ratsaby and S.S. Venkatesh

TEMPORAL DIFFERENCE LEARNING APPLIED TO MULTIHYPOTHESIS SEQUENTIAL DETECTION 1010
C. Guo and A. Kuh

A ROBUST TEMPORAL NEURAL NETWORK FOR BLIND CHANNEL IDENTIFICATION 1020
G. Dong and R.-w. Liu

INFORMATION THEORETIC LOWER BOUNDS ON CONVERGENCE RATES OF NONPARAMETRIC ESTIMATORS 1028
A. Barron and Y. Yang

SOME RESULTS ON NONPARAMETRIC REGRESSION ESTIMATION FOR ARBITRARY RANDOM PROCESSES 1033
S.E. Posner and S.R. Kulkarni

VI-E: MULTIUSER COMMUNICATIONS II

Organizer and Chairman: R. Pickholtz
George Washington University

ANALYSIS OF AN APPROXIMATE DECORRELATING DETECTOR 1043
N.B. Mandayam and S. Verdú

MULTIUSER DETECTION FOR FADING MULTIPATH MULTIUSER CHANNELS: A UNIFIED APPROACH TO DESIGN AND SUCCINCT EIGENBOUNDS ON PERFORMANCE: PART II.......... 1053
M.K. Varanasi

TWO-STAGE MULTIUSER DETECTION FOR NONCOHERENT CDMA.......... 1063
C. Hegarty and B. Vojcic

PERFORMANCE ANALYSIS OF THE DECORRELATING DETECTOR FOR DIRECT-SEQUENCE SPREAD-SPECTRUM MULTIPLE-ACCESS COMMUNICATION SYSTEMS 1073
I. Acar and S. Tantaratana

A CLOSED-LOOP JOINT PN PHASE AND CARRIER PHASE ACQUISITION SCHEME FOR DS/SS SYSTEMS.......... 1083
M. Salih and S. Tantaratana

AN ECONOMIC MODEL FOR UPLINK POWER CONTROL IN CELLULAR RADIO SYSTEMS..... 1093
H. Ji

INDEX OF AUTHORS.......... 1103

Combined Error Correction and Multiuser Detection for Rayleigh Fading Synchronous CDMA Channels [1]

Ayman El-Ezabi, Alexandra Duel-Hallen

North Carolina State University,
Department of Electrical and Computer Engineering,
Raleigh, NC 27695-7911
Phone/Fax: (919) 515-7352/5523
email: ael-eza@eos.ncsu.edu, sasha@eos.ncsu.edu

Abstract *Code Division Multiple Access (CDMA) signaling is a promising candidate for the emerging mobile and wireless communication systems standards. Present-day CDMA systems, however, employ single-user (conventional) detection which is far from optimum. While multiuser detectors offer improved capacity, little work has been reported on employing them in conjunction with error correction coding. We evaluate the improvements achieved by the use of channel coding for several classes of multiuser detectors over the Rayleigh fading channel, and compare alternative implementations for two-stage detectors.*

1 Introduction

There has recently been much interest in Code Division Multiple Access (CDMA) techniques as an alternative to narrowband signaling methods in wireless communications systems, mainly due to potential capacity increases and the fact that capacity can be increased at the expense of some gradual degradation in performance for all users.

The two main problems in the multiuser environment are Multiple Access Interference (MAI) and multipath fading. In the reverse link (from mobile to base station), the detector faces a challenging problem since user signals are asynchronous and the link suffers from the "near-far" problem, where weaker users are poorly detected.

The simplest receiver structure for the multiuser detection problem is the conventional (or single-user) detector. However, its performance is generally poor for bandwidth efficient systems and weaker users suffer greatly from the near-far problem, necessitating the use of strict power control to ensure that signals from the different users arrive at the base station with close to equal power levels. The Maximum-likelihood detector, on the other hand, offers optimum performance [1] at the expense of high complexity which becomes prohibitive even for systems with a moderate number of users. Several suboptimum detectors that attempt to offer a better performance/complexity ratio have therefore been proposed, e.g. [2, 3, 4, 5]. However, most of the research effort reported so far has considered uncoded systems over the Additive White Gaussian Noise (AWGN) channel with a few exceptions (e.g. [6, 7, 8]).

In this work, we compare several low to medium complexity multiuser detectors using coding over the Rayleigh fading channel for different noise and MAI conditions; namely the conventional detector, the decorrelator, and the two-stage detectors with the conventional and the decorrelating first stage (TSCFS and TSDFS, respectively). We also compare two alternative implementations of the two-stage detectors with channel coding. We use a synchronous model and rely on computer simulations in our analysis. The multiuser detectors considered here can be generalized to the asynchronous case [3], and therefore the results of our comparative performance analysis can be carried over to the asynchronous channel found in practice.

---

[1] **This research was supported by NSF grant NCR-9410227 and the Center for Advanced Computing and Communication at North Carolina State University**

In Section 2, we describe the fading channel model and the CDMA system. In Section 3, we describe different CDMA receiver structures and in Section 4 our simulation model. Section 5 presents the results of our comparisons, and in Section 6 we compare two alternative schemes for two-stage detectors with coding. Finally, Section 7 presents conclusions and possibilities for future work in the area.

2 CDMA Fading Channel Model

In a CDMA system, each user is assigned a unique signature sequence and all users transmit simultaneously over the same channel. Let $b_k(j)$ denote the jth information bit of user k which is drawn from $\{-1, 1\}$ for BPSK modulation. We assume a frequency-nonselective channel which leads to a ”multiplicative fading” expression for the received signal where the envelope for each user is multiplied by a realization, $c_k(t)$, of a zero-mean Gaussian complex random process. We consider a bit-synchronous system and since the information bits of each user are i.i.d., equiprobable and independent of other users', we may consider only one bit interval without loss of generality. We also assume slow fading in which case the amplitude and phase of $c_k(t)$ do not change significantly over one bit interval, and we may suppress the time-dependence of the fading coefficients for the duration of one bit. The received waveform is then given by

$$r(t) = \sum_{k=1}^{K} c_k b_k s_k(t) + n(t) \tag{1}$$

where $s_k(t)$ denotes the kth user's signature waveform, $n(t)$ is a realization of a zero-mean complex white Gaussian process with power spectral density σ^2, and c_k's are independent complex zero-mean Gaussian random variables. We consider normalized signature waveforms of duration T, i.e. $\int_0^T |s_i(t)|^2 dt = 1$.

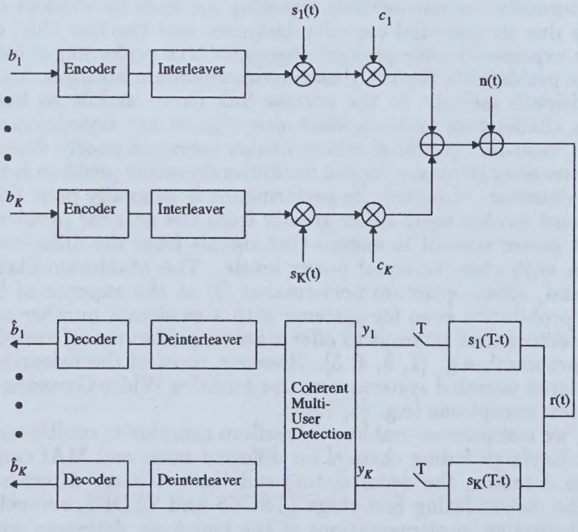


Figure 1: Uplink of a Coded CDMA system with fading

Figure 1 shows the uplink (from mobile to base station) of a coded CDMA system over a flat Rayleigh fading channel. The optimum filters at the receiver are a bank of filters matched to the signature waveforms of each user and the metrics at the output of the matched filters and samplers are given by

$$\mathbf{y} = \mathbf{RCb} + \mathbf{n} \tag{2}$$

where $\mathbf{C}=\mathrm{diag}\{c_1, ..., c_K\}$ is the complex path gain matrix, and $\mathbf{n}$ is a zero-mean complex Gaussian $K-$vector with the covariance matrix $\sigma^2\mathbf{R}$ for both real and imaginary components. $\mathbf{R}$ is the matrix of cross-correlations between the normalized signature waveforms with elements :

$$R_{ij} = \int_0^T s_i(t)s_j(t)\ dt \tag{3}$$

In our comparative analysis, we consider the conventional, decorrelating, and two-stage detectors.

3 Multiuser Receiver Structures

The simplest detector is the conventional (or single-user) detector. It ignores Multiple Access Interference (MAI) and applies the decision rule

$$\hat{\mathbf{b}} = \mathbf{sgn}\ \grave{\mathbf{y}} \tag{4}$$

where

$$\grave{\mathbf{y}} = \mathbf{Re}[\mathbf{C}^*\mathbf{y}] = \mathbf{Re}[\mathbf{C}^*\mathbf{RC}]\mathbf{b} + \mathbf{Re}[\mathbf{C}^*\mathbf{n}] \tag{5}$$

The decorrelator [2] is a linear detector which eliminates MAI at the expense of increased noise power. The decorrelating matrix, $\mathbf{R}^{-1}$, is applied to the output of the matched filter bank :

$$\tilde{\mathbf{y}} = \mathbf{R}^{-1}\mathbf{y} = \mathbf{R}^{-1}[\mathbf{RCb} + \mathbf{n}] = \mathbf{Cb} + \tilde{\mathbf{n}} \tag{6}$$

where $\tilde{\mathbf{n}}$ is a zero-mean complex Gaussian $K-$vector with the covariance matrix $\sigma^2\mathbf{R}^{-1}$. The inputs to the threshold devices are the components of the vector

$$\grave{\tilde{\mathbf{y}}} = \mathbf{Re}[\mathbf{C}^*\tilde{\mathbf{y}}] = \mathbf{Re}[\mathbf{C}^*(\mathbf{Cb} + \tilde{\mathbf{n}})] = |\mathbf{C}|^2\mathbf{b} + \mathbf{Re}[\mathbf{C}^*\tilde{\mathbf{n}}]\ , \tag{7}$$

and the decision rule is

$$\hat{\mathbf{b}} = \mathbf{sgn}\ \grave{\tilde{\mathbf{y}}} \tag{8}$$

Multistage detectors consist of an initial stage where tentative decisions are made for all users followed by one or more stages where more reliable decisions are obtained. While multi-stage detectors are non-linear detectors due to MAI cancelation that occurs in subsequent stages, they have linear complexity in the number of users. We consider the two-stage detector proposed in [3], with conventional and decorrelating first stages. The bit estimates of the first stage are denoted by $\hat{\mathbf{b}}(1) = [\hat{b}_1(1), ..., \hat{b}_K(1)]$. Denoting the matrix $\{\mathrm{Re}[\mathbf{C}^*\mathbf{RC}]\}$ by $\mathbf{M}$, the second stage bit estimate for the kth user is given by :

$$\hat{b}_k(2) = \mathrm{sgn}\left[\grave{\mathrm{y}}_{\mathrm{k}} - \sum_{\mathrm{j}\neq\mathrm{k}} \hat{\mathrm{b}}_{\mathrm{j}}(1)\mathrm{M}_{\mathrm{jk}}\right] \tag{9}$$

where $\{\grave{y}\}_k$ are the elements of the vector $\grave{\mathbf{y}}$ given by Equation 5 and the first stage decisions are given by either of the conventional or decorrelating detectors.

Figure 2 shows a simplified diagram of a two-stage detector with arbitrary first stage.

4 The Simulation Model

Since closed form expressions for the probabilities of error in coded systems are very difficult to obtain (even some of the uncoded systems we study are hard to analyze, e.g. multistage detectors), we rely on computer simulations to compare the performance of the different detectors.

We study a 4-user bit-synchronous system with a set of signature waveforms derived from Gold sequences of length seven as used e.g. in [3, 4]. These signature waveforms have the property that the cross-correlation between the signals is as low as possible

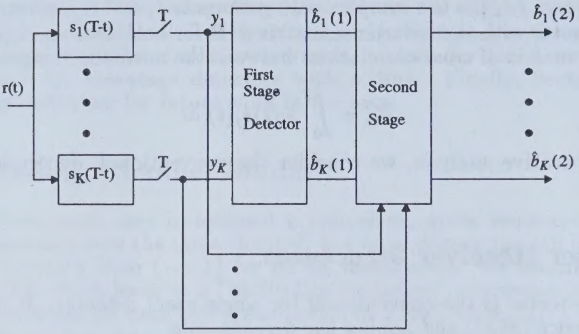


Figure 2: Two-stage Detector with Arbitrary First Stage

(given a certain allowable bandwidth) for all relative delays [3]. Since practical systems are asynchronous in nature, this is very important for drawing meaningful conclusions from our synchronous model. The resulting normalized cross-correlation matrix is [4]

$$\mathbf{R} = \frac{1}{7}\begin{bmatrix} 7 & -1 & 3 & 3 \\ -1 & 7 & -1 & 3 \\ 3 & -1 & 7 & -1 \\ 3 & 3 & -1 & 7 \end{bmatrix} \tag{10}$$

We assume BPSK modulation with coherent detection. We make the usual assumption of perfect channel estimation, and perfect interleaving so that the fading coefficients, c_k, are uncorrelated from one bit interval to another. We perform our simulation runs between 10^6 and 10^7 times, depending on the estimated bit error rate. For a number of runs equal to $100/\hat{P}_e$, and for a confidence level of 95%, we obtain approximate confidence intervals equal to $(1.216\hat{P}_e, 0.822\hat{P}_e)$ calculated from Equation (9) of [9], where $\hat{P}_e$ is the estimated probability of error.

For the coded channel, we use the half-rate four-state convolutional encoder with generator sequences $\mathbf{g}_1 = [101]$ and $\mathbf{g}_2 = [111]$. It has a minimum free distance, d_{free}, of 5 and its encoder memory, m, is equal to 2. We chose this encoder since it has a small decoding delay and its minimum free distance ($d_{\text{free}} = 5$) is the largest possible for an $R = (1/2)$ code with $m = 2$ ([10], Table 11.1).

At the receiver, we implement a hard-decision truncated best-state Viterbi decoder with a path memory of 10 branches. The truncation depth was chosen to be five times the encoder memory, which results in a maximum "truncation loss" of about 0.05 dB on the AWGN channel [11]. The above confidence intervals assume independent errors on the channel. We have error dependence due to encoding, however, which extends over 10 information bits. We obtain the same confidence intervals and confidence level, therefore, for a number of runs equal to $1000/\hat{P}_e$.

5 Comparison of Multiuser Detectors

Figures 3 and 4 show the BER plots versus average SNR for single-user and multiuser detectors for the uncoded and coded systems, respectively. The multiuser detectors generally outperform the conventional detector by several orders of magnitude of the BER. The TSDFS provides practically single-user performance over both the uncoded and coded channels. Table 1 gives the reduction in BER due to channel coding for the different detectors at different SNR points, and also the power gain in dB at a BER of 10^{-3}.

Table 1: Comparison of MU Detectors for Coded and Uncoded Systems on Fading Channel

	BER Reduction Factor (Orders of Magnitude)			Coding Gain @ 10^{-3} BER (dB)
	@13 dB	@18 dB	@23 dB	
TSCFS	1	≤ 2	2 to 3	NA
Decorrelator	0 to 1	1 to 2	2 to 3	11
TSDFS	1	≥ 2	3	12
SUB	1 to 2	≥ 2	3	11.5

We notice that the conventional detector's performance improves very little with coding and with increasing SNR. Figure 5 shows the performance of the conventional-based detectors for a system with the normalized cross-correlation matrix given by

$$\mathbf{R} = \begin{bmatrix} 1.0 & 0.2 & 0.2 & 0.2 \\ 0.2 & 1.0 & 0.2 & 0.2 \\ 0.2 & 0.2 & 1.0 & 0.2 \\ 0.2 & 0.2 & 0.2 & 1.0 \end{bmatrix} \tag{11}$$

We find that the conventional detector still has a relatively flat characteristic for SNR values beyond 10 and 15 dB for the uncoded and coded cases, respectively, resulting in a relatively high error floor.

On the other hand, returning to Figures 3 and 4, we find that the TSCFS detector's performance improves greatly with coding, and it's BER drops to about 10^{-6} before approaching the error floor. It also outperforms the decorrelator up to an average SNR value of 23 dB. The main advantage of the TSCFS detector (and non-decorrelating detectors generally) over the decorrelator is that no matrix inversion is required at the receiver. We also find that the TSCFS detector without coding does not approach its error floor over the SNR range of interest for the lower bandwidth-efficiency system of Figure 5.

We now study the important near-far situation where user 1's energy is, on average, equal to half the energies of the other three users. As expected, the conventional detector's BER is extremely high. User 1's performance is unaffected when the decorrelator is used since its performance is independent of other users' energies. Figure 6 shows the performance of two-stage detectors with conventional and decorrelating first stages for uncoded and coded systems. We observe that the TSCFS detector's performance is slightly worse than in the average equal SNR case. The change in performance is due to two opposing effects. Unlike in the 2-user case where the weaker user benefits greatly from MAI cancelation, the three stronger (on average) users here still interfere significantly with each other. Thus, they have only slightly better first stage estimates due to their higher energies and weaker interference from user 1. On the other hand, the metric at the output of the matched filter for user 1, i.e. y_1, now has higher MAI terms, which results in a slight overall increase in BER for user 1. We find that this is also the case for the AWGN channel for both the uncoded and coded systems.

6 Two-Stage Detector Implementations for Coded Systems

An alternative, and more complex, implementation (we refer to it as the complex scheme) than the one described in Figure 1 (referred to as the simple scheme) for two-stage detectors on a fading channel is shown in Figure 7. Here, error correction is applied to the decisions of the first stage detector. The decoding delay of this method is twice that of the simpler implementation of Figure 1.

While it may be expected that the complex scheme would outperform the simple scheme for all SNR values due to better first stage estimates (the only exception being when the BER is so high that the decoder introduces more errors), we find that it is not the case. For the additive white Gaussian noise channel, the superiority of a particular scheme depends on the average SNR and the cross-correlation between the systems users and the simple scheme is always at least as good as the complex scheme for the fading channel (Figure 8).

One explanation for this behavior is that the decoding applied to the first stage decisions introduces error dependence into the bit sequences entering the second bank of Viterbi decoders, and the channel ceases to be memoryless from that point onwards. Thus, the Viterbi decoders no longer provide optimum decisions. This effect is offset, however, by the improved decisions of the first stage and the complex scheme outperforms the simple scheme for low SNR values. The simple scheme always outperforms the complex scheme for higher SNR since the decoding of first stage decisions becomes less critical to subsequent stages, and the non-optimality of the second Viterbi decoder bank in the complex scheme results in a net degradation in performance.

The cross-over point between both schemes (if it exists) also depends on the cross-correlations between the system users. When we compare both schemes for a lower bandwidth-efficiency channel (**R** given by Equation 11), we find both schemes have equal performance, as shown in Figure 5. Moreover, for the AWGN channel, where MAI is less severe, the complex scheme outperforms the simple scheme for low SNR values when **R** is given by Equation 10.

The reason for this behavior is that for low cross-correlations between system users, the MAI interference cancelation capability of the second stage is very limited. Thus, the "changes" introduced by the second stage to the path chosen by the first decoder bank are very few and the non-optimality of the second decoder bank is insignificant. In addition, for low SNR values the advantage of improved first stage decisions being fed into the second stage results in better overall performance for the complex scheme. For the same level of cross-correlations between system users, the simple scheme does better than the complex scheme on the fading channel since it is inherently a "near-far" channel.

7 Conclusions

In this work, we evaluated the performance gap between single- and multi-user detectors for typical average SNR values on a flat fading channel. We find that with a simple half-rate code, two-stage detectors outperform the conventional detector by up to five orders of magnitude of the BER and that lower cross-correlations between system users do not significantly improve the conventional detector's performance. The TSCFS detector, on the other hand, improves greatly with lower user cross-correlations. For the coded system, it outperforms the decorrelator for up to a SNR value of 23 dB, and its BER for larger SNR values is almost 10^{-6}. Moreover, its performance in a simulated near-far situation is almost unaffected. The crucial advantage of the TSCFS detector over the decorrelator is that is does not require cross-correlation matrix inversion at the receiver.

We compare two alternative schemes (simple and complex) for employing error correction with two-stage detectors and analyze their performance. Under investigation is the use of concatenated coding to avoid the non-optimal decisions of the second Viterbi decoder bank in the complex scheme. We are also planning to study performance of simple multiuser detectors such as the TSCFS detector and interference cancelation method ([12]) for realistic Rayleigh fading channels.

References

[1] S. Verdu. Minimum Probability of Error for Asynchronous Gaussian Multiple-Access Channels. *IEEE Transactions on Information Theory*, IT-32(1):85–96, Jan. 1986.

[2] R. Lupas and S. Verdu. Linear Multiuser Detectors for Synchronous Code-Division Multiple-Access Channels. *IEEE Transactions on Information Theory*, IT-35(1):123–136, Jan. 1989.

[3] M. K. Varanasi and B. Aazhang. Near-Optimum Detection in Synchronous Code-Division Multiple-Access Systems. *IEEE Transactions on Communications*, COM-39(5):725–736, May 1991.

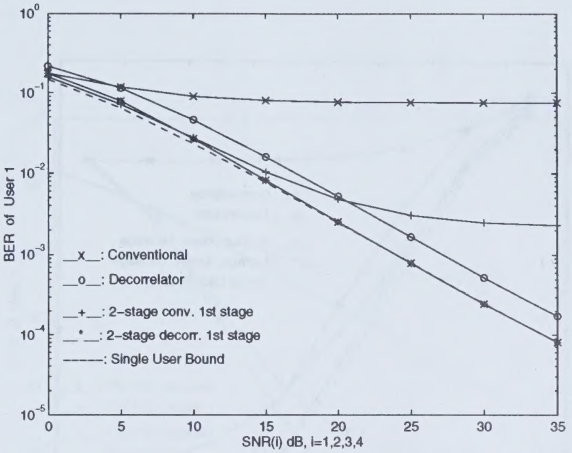


Figure 3: BER for Uncoded 4-user Systems on a Rayleigh Fading Channel

[4] A. Duel-Hallen. Decorrelating Decision-Feedback Multiuser Detector for Synchronous Code-Division Multiple-Access Channel. *IEEE Transactions on Communications*, COM-41(2):285–290, Feb. 1993.

[5] Zhenhua Xie, Robert Short, and Craig Rushforth. A Family of Suboptimum Detectors for Coherent Multiuser Communications. *IEEE Journal on Selected Areas in Communications*, 8(4):683–690, May 1990.

[6] Peter Hoeher. On Channel Coding and Multiuser Detection for DS-CDMA. In *Proceedings of the 2nd International Conference on Universal Personal Communications, Ottawa, Canada*, pages 641–646, October 1993.

[7] T. R. Giallorenzi. Suboptimum Multiuser Receivers for Convolutionally Coded Asynchronous CDMA Systems. Preprint.

[8] M. Nasiri-Kenari and C.K. Rushforth. An Efficient Soft-Decision Decoding Algorithm for Synchronous CDMA with Error-Control Coding. In *Proceedings of the IEEE International Symposium on Information Theory, Trondheim, Norway*, page 227, June 1994.

[9] Michel C. Jeruchim. Techniques for Estimating the Bit Error Rate in the Simulation of Digital Communication Systems. *IEEE Journal on Selected Areas in Communications*, 2(1):153–170, January 1984.

[10] S. Lin and D.J. Costello, Jr. *Error Control Coding - Fundamentals and Applications.* Prentice-Hall, Englewood Cliffs,NJ 07632, 1983.

[11] I.M. Onyszchuk. Truncation Length for Viterbi Decoding. *IEEE Transactions on Communications*, COM-39(7):1023–1026, July 1991.

[12] Pulin Patel and Jack Holtzman. Analysis of a Simple Successive Interference Cancellation Scheme in a DS/CDMA System. *IEEE Journal on Selected Areas in Communications*, 12(5):796–807, June 1994.

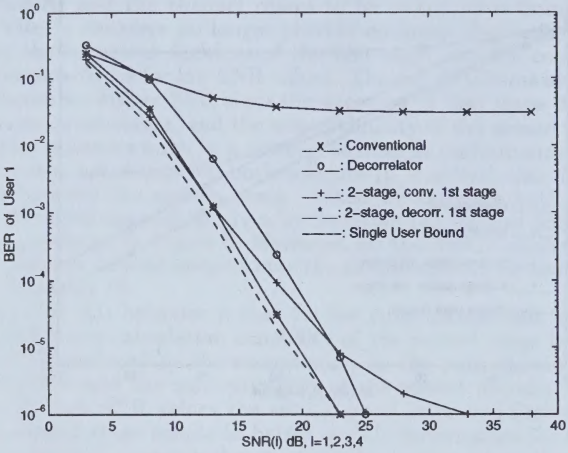


Figure 4: BER for Coded 4-user Systems on a Rayleigh Fading Channel

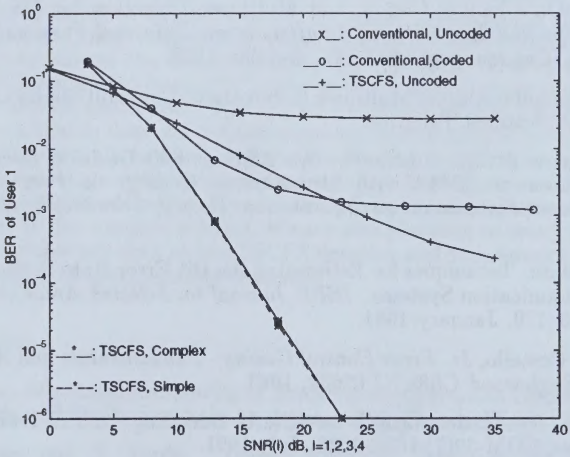


Figure 5: BER for Conventional-based Detection on a Lower Bandwidth-efficiency System

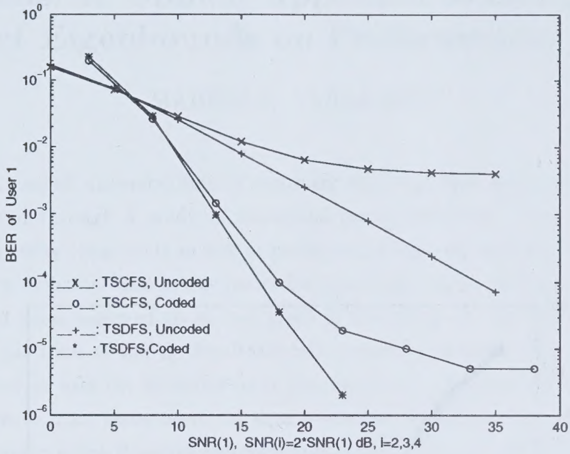


Figure 6: BER for Two-stage Detectors in a near-far environment

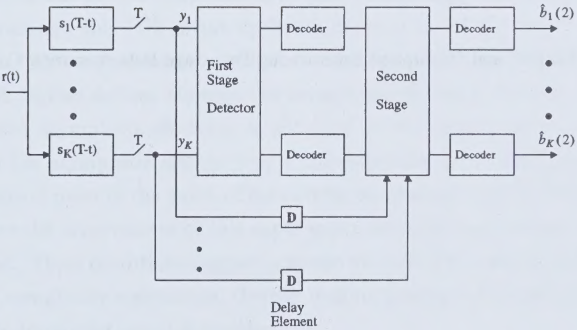

r(t)
s1(T-t)
sK(T-t)
T
T
y1
yK
First
Stage
Detector
Decoder
Decoder
Second
Stage
Decoder
Decoder
b̂1(2)
b̂K(2)
D
D
Delay
Element

Figure 7: Complex Implementation of a Two-stage Detector with Coding

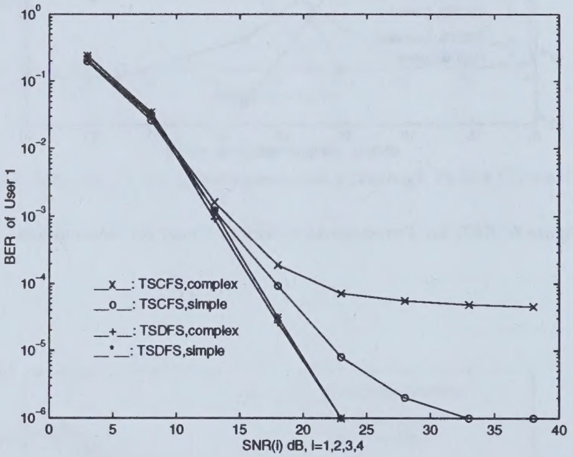


Figure 8: BER of "Simple" and "Complex" Schemes for Two-stage Detectors with Coding in a Fading Channel

Multiuser Detection for Fading Multipath Multiuser Channels: A Unified Approach to Design and Succinct Eigenbounds on Performance: Part I

MAHESH K. VARANASI†

Summary: A unified understanding of multiuser detection over fading multipath multiuser channels is achieved through a study of sequential group detection. This includes optimum design under complexity constraints as well as performance analysis. A sequential group detector consists of J group detectors that are connected sequentially, where each group detector jointly detects a group of users assigned to it, and there is one group detector for each group of an ordered J group partition of the K simultaneously transmitting users of a multiuser channel. The j^{th} group detector uses the decisions from the previous $j-1$ group detectors to cancel the inter-user interference from those users. It then tunes out the interfering signals from users in future groups following which it makes optimum joint decisions on the symbols of users in its own group. In this paper, the first of two parts, we consider *post-combining sequential group detection*, where the signals from users in future groups are tuned out by projecting the residual received signal that remains after subtracting the interference from past groups, onto the orthogonal complement of the signal subspace spanned by the interfering signals in future groups and their multipath-induced time-translates. Sequential group detection allows the strong users to come to the aid of the weak users by parsimoniously sacrificing their own performance which they are presumably better able to withstand. The Time Complexity per Symbol (TCS) of the proposed post-combining sequential group detector is $O(\sum_{j=1}^{J} M^{|G_j|})$ for M-ary signalling with $|G_j|$ being the j^{th} group size. It can therefore be controlled over a wide range by an appropriate choice of group sizes. In contrast, the TCS of the optimum detector is $O(\prod_{j=1}^{J} M^{|G_j|})$ and is therefore fixed and too high. As for performance analysis, a succinct indicator of the average BER over medium-high SNR regions defined through the asymptotic efficiency measure is considered. An upper bound on the asymptotic efficiency is obtained for the sequential group detector. This bound is equal to the asymptotic efficiency of a non-sequential super-group detector where the super-group consists of users in the union of the current and past groups. Sufficient conditions are derived that ensure the achievement of this super-group detector bound when error propagation effects are included. These conditions suggest concrete methods for selecting ordered partitions of users under given complexity constraints, thereby making precise the folkloristic rule of ordering users according to decreasing signal strengths.

---

† The author is with the department of Electrical and Computer Engineering at the University of Colorado, Boulder, CO 80309. This work was supported by NSF grant NCR-9406069.

1 System Model

The received signal is modeled as a superposition of K digitally modulated waveforms that arrive at the receiver simultaneously after having passed through *slowly fading* frequency-selective Rayleigh fading channels. The coherence time of such channels is much greater than the data symbol duration so that accurate measurements of the channel characteristics are possible. Furthermore, the bandwidth assigned to each of the signature waveforms W is assumed to be much larger than the coherence bandwidth B_k of the k^{th} channel so that each channel can be represented as a tapped delay line with tap spacing $1/W$, and tap weight functions $\{c_{k,i}(t)\}_{i=0}^{L_k-1}$ [1]. These fading parameters represent the channel state information and are modeled as zero-mean complex-valued stationary Gaussian random processes. The number of resolvable paths L_k for the k^{th} signal is $\lceil W/B_k \rceil$, where $1/B_k$ is the k^{th} channel's multipath spread. The k^{th} user's signature waveform (possibly complex-valued) is time-limited to a symbol duration T, and is denoted as $u_k(t)$. The complex baseband representation of the signal that arrives at the receiver due to the isolated transmission of the modulated signature waveform of the k^{th} user is given as

$$\sqrt{w_k} \sum_{i=0}^{L_k-1} c_{k,i}(t) u_k \left(t - \frac{i}{W} \right). \tag{1}$$

When a sequence of information symbols is transmitted, there is intersymbol interference (ISI). In wideband single-user communications where the symbol duration is much larger than the multipath spread, so that $L_k/W << T$, the ISI due to channel dispersion is usually ignored (cf. [1], [2]), thereby facilitating symbol-by-symbol decisions. In wideband synchronous CDMA communications however, there is ISI as well as inter-user interference (IUI), and even though ISI and IUI occur over a small fraction of a single symbol interval, the ever-present possibility of near-far conditions may accentuate their harmful effect, if ignored. Therefore, we choose to eliminate rather than ignore these interferences by masking out the received signal over time-intervals where both ISI and IUI are present [3]. The small loss in optimality is thereby traded for the simplicity of making symbol-by-symbol decisions. The complex baseband representation of the received signal over the thusly restricted zeroth time-interval is given by

$$r(t) = \sum_{k=1}^{K} \sum_{i=0}^{L_k-1} \sqrt{w_k} b_k c_{k,i}(t) u_k \left(t - \frac{i}{W} \right) + n(t) , \qquad (L_M - 1)/W \leq t \leq T \tag{2}$$

where $L_M \triangleq \max L_k$ and $n(t)$ denotes a circularly symmetric, complex, additive, and white, Gaussian noise process with a power spectral density of height N_0. The symbol b_k belongs to F_k, an M-ary complex-valued symbol constellation of the k^{th} user.

A succinct description of the problem is facilitated by using the following matrix notation. Let $\mathbf{b} = [b_1, \cdots, b_K]^T$ with b_k denoting the k^{th} user's information symbol. Let the k^{th} fading channel vector be defined as $\mathbf{c}_k(t) = [c_{k,0}(t), \cdots, c_{k,L_k-1}(t)]^T$, and the k^{th} user's signature waveform and its time-translates be defined by the vector $\mathbf{u}_k(t) = [u_k(t) \ u_k(t - 1/W) \ \cdots \ u_k(t - (L_k - 1)/W)]^T$. The L-length column vector (with $L \triangleq \sum_{k=1}^{K} L_k$) of signals obtained by stacking the vectors

$\{\mathbf{u}_k(t)\}_{k=1}^K$ is denoted as $\mathbf{u}(t)$. The $L \times L$ signature signal correlation matrix is thus defined as

$$\mathcal{R} = \int_{(L_M-1)/W}^{T} \mathbf{u}^*(t)\mathbf{u}^T(t)dt = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} & \cdots & \mathbf{R}_{1K} \\ \mathbf{R}_{21} & \mathbf{R}_{22} & \cdots & \mathbf{R}_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{K1} & \mathbf{R}_{K2} & \cdots & \mathbf{R}_{KK} \end{bmatrix}, \tag{3}$$

with the superscripts $*$ and T denoting complex conjugation and matrix transposition, respectively. The matrix $\mathbf{R}_{kl}$ is the correlation matrix between the L_k and the L_l-dimensional signal vectors $\mathbf{u}_k(t)$ and $\mathbf{u}_l(t)$, respectively, defined as $\mathbf{R}_{kl} = \int_{(L_M-1)/W}^{T} \mathbf{u}_k^*(t)\mathbf{u}_l^T(t)dt$. The signature signals are normalized to have unit energy and all their time-translates are assumed to be linearly independent so that $\mathcal{R}$ is positive definite. The diagonal matrix $\mathbf{W} = diag\{w_1, \cdots, w_K\}$ is such that w_k is the k^{th} user's total average received signal power, so that the expected value of $\int_{(L_M-1)/W}^{T} |\mathbf{c}_k^T(t)\mathbf{u}_k(t)|^2 dt = 1$. Let the covariance matrix of $\mathbf{c}_k$ be denoted as $\mathbf{\Sigma}_{kk}$. It is also convenient to define the $L \times K$ multipath multiuser channel state information matrix $\mathcal{C}$ as

$$\mathcal{C} = \begin{bmatrix} \mathbf{c}_1(t) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{c}_2(t) & \cdots & \mathbf{0} \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{c}_K(t) \end{bmatrix}, \tag{4}$$

where it is assumed that this matrix is essentially constant over a single symbol duration.

Let $\mathbf{q}$ denote the L-dimensional sufficient statistic vector that is obtained by stacking the L_k-dimensional vectors $\{\mathbf{q}_k\}_{k=1}^K$, where $\mathbf{q}_k$ is defined as

$$\mathbf{q}_k = \int_{(L_M-1)/W}^{T} r(t)\mathbf{u}_k^*(t)\, dt\,. \tag{5}$$

2 Sequential Group Detection

Notation: Consider an ordered partition of the K users according to $\Omega = \{1, 2, \cdots, K\} = \cup_{j=1}^J G_j$. Without loss of generality, we let $G_j = \{K_{j-1}+1, \cdots, K_j\}$ with $K_0 = 0$ and $K_J = K$. Any arbitrary ordered partition of the K users can be expressed this way by an appropriate renumbering of the users so that users intended for group G_j would be indexed from $K_{j-1}+1$ to K_j for $1 \leq j \leq J$.

Consider the K-dimensional vector of symbols $\mathbf{b}$. We define several useful partitions of $\mathbf{b}$ that are induced by the ordered partition of users defined above. Let $\mathbf{b}^l$ be the vector of symbols of users in group G_l. Let $\mathbf{b}^{(m,j)}$ be obtained by stacking the vectors $\mathbf{b}_l$ for l ranging consecutively from m to j. Partitions of any vector that has the dimension and structure of $\mathbf{b}$ are defined similarly. Next, we define partitions of the vector of sufficient statistic vector $\mathbf{q}$. Let $\mathbf{q}_k$ be the k^{th} vector of length L_k. Let $\mathbf{q}^m$ be the vector obtained by stacking $\mathbf{q}_i$ for all $i \in G_m$. Let $\mathbf{q}^{(m,j)}$ be the vector obtained by stacking $\mathbf{q}^l$ for l ranging consecutively from m to j. Partitions of any vector that has the dimension and structure of $\mathbf{q}$ are defined similarly.

The diagonal matrix of average received signal powers $\mathbf{W}$ admits the obvious partition in terms of the group signal power matrices $\{\mathbf{W}^j\}_{j=1}^J$ as $\mathbf{W} = \mathrm{diag}\{\mathbf{W}^1, \cdots, \mathbf{W}^J\}$ and we define the submatrix $\mathbf{W}^{(l,k)} = \mathrm{diag}\{\mathbf{W}^l, \cdots, \mathbf{W}^k\}$. It is convenient to express the block-diagonal channel state matrix $\mathcal{C}$ in terms of the channel state matrices of each of the J groups so that $\mathcal{C} = \mathrm{diag}\{\mathcal{C}^1, \mathcal{C}^2, \cdots \mathcal{C}^J\}$ with the block-diagonal $\mathcal{C}^j$ containing the channel state vectors of users in group G_j as its diagonal elements. Furthermore, define the diagonal block submatrix of $\mathcal{C}$ formed from $\mathcal{C}^m$ to $\mathcal{C}^j$ as $\mathcal{C}^{(m,j)} = \mathrm{diag}\{\mathcal{C}^m \cdots \mathcal{C}^j\}$. Moreover, partitions of arbitrary matrices with the dimension and structure of $\mathbf{W}$ or $\mathcal{C}$ are defined similarly.

It is convenient to partition the correlation matrix $\mathcal{R}$ as

$$\mathcal{R} = \begin{bmatrix} \mathcal{R}^{11} & \mathcal{R}^{12} & \cdots & \mathcal{R}^{1J} \\ \mathcal{R}^{21} & \mathcal{R}^{22} & \cdots & \mathcal{R}^{2J} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{R}^{J1} & \mathcal{R}^{J2} & \cdots & \mathcal{R}^{JJ} \end{bmatrix}, \tag{6}$$

where the $\sum_{l \in G_i} L_l \times \sum_{l \in G_j} L_l$ matrix $\mathcal{R}^{mj}$ is the crosscorrelation matrix between signals of groups G_m and G_j. Furthermore, define the submatrix of $\mathcal{R}$ obtained by extracting group block rows consecutively from m to l and group block columns n to j as $\mathcal{R}^{(m,l)(n,j)}$ so that

$$\mathcal{R}^{(m,l)(n,j)} = \begin{bmatrix} \mathcal{R}^{mn} & \cdots & \mathcal{R}^{mj} \\ \vdots & \ddots & \vdots \\ \mathcal{R}^{ln} & \cdots & \mathcal{R}^{lj} \end{bmatrix}. \tag{7}$$

Finally, we note that partitions of matrices with the structure and dimension of $\mathcal{R}$ are defined similarly. We are now ready to describe the post-combining sequential group detector.

POST-COMBINING SEQUENTIAL GROUP DETECTION

The first step involves the post-combining group detection of users in the first group. The second and the third steps describe the interference cancellation process followed by the group detection operation. These steps are performed successively starting with $j = 2$ and terminating at $j = J$.

1. *Group detection for the first group:* An optimum detection of symbols that includes the Group Maximal Ratio Combining (GMRC) operation of the users in the first group G_1 is obtained after projecting the received signal onto the orthogonal complement of the subspace spanned by the signature waveforms of the users in future groups $G_2, \cdots, G_J$ and their multipath-induced time-translates [3]. This rule is denoted as ϕ_{G_1} and is described by

$$\begin{aligned} \phi_{G_1}: \quad \hat{\mathbf{b}}^1 = \arg \max_{\mathbf{x} \in F^1} \quad & 2 \quad \mathrm{Re}\left(\mathbf{x}^{*T} \left[\mathbf{W}^1\right]^{1/2} \mathcal{C}^{1*T} \mathcal{P}^{11} \left[\mathcal{R}^{-1}\mathbf{q}\right]^1 \right) \\ & - \quad \mathbf{x}^{*T} \left[\mathbf{W}^1\right]^{1/2} \mathcal{C}^{1*T} \mathcal{P}^{11} \mathcal{C}^1 \left[\mathbf{W}^1\right]^{1/2} \mathbf{x}, \end{aligned} \tag{8}$$

   where $\mathcal{P}^{11} \triangleq \left[[\mathcal{R}^{-1}]^{11}\right]^{-1}$ and F^1 denotes the joint alphabet of the symbols $\mathbf{b}^1$.

2. *Interference Cancellation:* The multiuser interference contributed by users in the previously detected groups $G_1, \cdots, G_{j-1}$ to the matched filter output vectors of users in the current and future groups $G_j, \cdots, G_J$ is estimated and subtracted,
$$\mathbf{z}^{(j)} = \mathbf{q}^{(j,J)} - \mathcal{R}^{(j,J)(1,j-1)}\mathcal{C}^{(1,j-1)} \left[\mathbf{W}^{(1,j-1)}\right]^{1/2} \hat{\mathbf{b}}^{(1,j-1)} . \tag{9}$$
With the assumption of perfect interference cancellation, there is no loss of sufficiency in restricting attention to $\mathbf{q}^{(j,J)}$.

3. *Sequential Group Detection for G_j:* The symbols of users in group G_j are detected optimally after tuning out the interfering signals and their time-translates from future groups $G_{j+1}, \cdots, G_J$. This is accomplished by projecting the residual received signal that remains after subtracting the interference from past groups, onto the orthogonal complement of the subspace spanned by the users in the future groups. This is accomplished by the j^{th} sequential group detector which is denoted as $\tilde{\phi}_{G_j}$ and is defined as
$$\begin{aligned}\tilde{\phi}_{G_j}: \quad \hat{\mathbf{b}}^j = \arg\max_{\mathbf{x}\in F^j} \quad 2 \quad & \mathrm{Re}\left(\mathbf{x}^{*T}\left[\mathbf{W}^j\right]^{1/2} \mathcal{C}^{j*T}\mathcal{P}^{jj}\left[\mathcal{Q}^{(j)}\mathbf{z}^{(j)}\right]^1\right) \\ - \quad & \mathbf{x}^{*T}\left[\mathbf{W}^j\right]^{1/2}\mathcal{C}^{j*T}\mathcal{P}^{jj}\mathcal{C}^j\left[\mathbf{W}^j\right]^{1/2}\mathbf{x} ,\end{aligned} \tag{10}$$
where F^j is the joint alphabet of the symbols $\mathbf{b}^j$, and where we let $\mathcal{Q}^{(j)} = \left[\mathcal{R}^{(j,J)(j,J)}\right]^{-1}$. Furthermore, we define $\mathcal{P}^{jj}$ as the inverse of $\left[\mathcal{Q}^{(j)}\right]^{(1,1)}$ which is the first diagonal block of dimension $\sum_{l\in G_j} L_l \times \sum_{l\in G_j} L_l$ of the block matrix representation of $\mathcal{Q}^{(j)}$. Also, $\left[\mathcal{Q}^{(j)}\mathbf{z}^{(j)}\right]^{(1)}$ represents the first subvector of dimension $\sum_{l\in G_j} L_l$ in the natural block vector representation of $\mathcal{Q}^{(j)}\mathbf{z}^{(j)}$.

The complexity of the above sequential group detector is determined by its TCS which is given as $O(\sum_{j=1}^{J} M^{|G_j|})$ for M-ary signalling with $|G_j|$ being the j^{th} group size. The particular case of the sequential group detector when all group sizes are equal to unity will be called the decorrelating-combiner based decision feedback detector. It can as such be regarded as a generalization from the Gaussian to the fading multipath multiuser channel of the decorrelating decision feedback detector (cf. [4] [5]). Moreover, the sequential group detector yields the optimum detector [6] for the trivial partition where all users are in one group of size K.

3 Conditional Asymptotic Efficiency

From a detector design viewpoint, in applications where the channel state information matrix $\mathcal{C}$ is acquired at the receiver, one ideally seeks optimality in a measure of performance that is conditioned on this information. In what follows, we define the conditional asymptotic efficiency measure, and through it, we establish a minimax optimality of the sequential group detector.

Definition: The *conditional asymptotic efficiency* of any user k employing the detector ϕ whose conditional bit error probability for that user is $P_k(\mathbf{w}, N_0, \phi/\mathcal{C})$, is formally defined as

$$\eta_k(\phi/\mathcal{C}) \triangleq \sup\left\{ 0 \le r \le 1;\ \ \lim_{N_0 \to 0} \ \frac{P_k(\mathbf{w}, N_0, \phi/\mathcal{C})}{Q\left(\sqrt{\frac{2 r w_k \mathbf{c}_k^{*T} \mathbf{R}_{kk} \mathbf{c}_k}{N_0}}\right)} < \infty \right\}, \tag{11}$$

with the denominator being the minimum conditional bit error probability achievable in a single user channel where the k^{th} user transmits in isolation and has an average received signal energy per symbol that is equal to $r w_k$.

Proposition 1 *Under the assumption of perfect multiuser interference cancellation, the sequential group detector* $\tilde{\phi}_{G_j}$ *has the highest worst-case conditional asymptotic efficiency for every user in* G_j*—with the worst-case being taken over the signal powers and amplitudes of all paths of the users in future groups* $G_{j+1}, \cdots, G_J$*—that is achievable by any detector that has perfect side-information about symbols of users in previous groups* $G_1, \cdots, G_{j-1}$ *(this class of genie-aided detectors is denoted as* Φ_j*) which is the minimax optimality result:*

$$\phi_{G_j} \in \arg\max_{\phi \in \Phi_j} \ \inf_{\mathcal{C}^{(j+1,J)}} \ \inf_{\mathbf{W}^{(j+1,J)}} \ \eta_k(\phi/\mathcal{C}), \tag{12}$$

where max *denotes element-wise maximum, i.e., the sequential group detector maximizes the worst-case conditional asymptotic efficiency for every user in group* G_j*.*

From a detector design point of view, the conditional nature of the above performance measure is ideal–since the optimality in this measure is assured for every realization of $\mathcal{C}$–provided of course that the ordered partition can be chosen such that the assumption of perfect cancellation is in fact valid. From an analysis view-point however, it must be noted that the expected asymptotic efficiency, defined as the expected value of the conditional asymptotic efficiency, may not be a faithful measure of the expected bit-error rate.

4 Asymptotic Efficiency

Let the average bit error probability of a detector ϕ for the k^{th} user in a FSRF-CDMA channel be denoted as $P_k(\mathbf{w}, N_0, \phi)$, where $\mathbf{w}$ is the vector of average received signal power of the K users. Let $P_k^{SU}(w, N_0)$ be the average bit-error rate of the single-user RAKE receiver operating in the k^{th} single-user FSRF channel where the k^{th} user transmits in isolation and whose signal at the receiver has average signal power equal to w.

Definition: The *asymptotic efficiency* of a detector ϕ operating in the FSRF-CDMA channel is formally defined as

$$\eta_k(\phi) = \sup\left\{ 0 \le r \le 1;\ \ \lim_{N_0 \to 0} \ \frac{P_k(\mathbf{w}, N_0, \phi)}{P_{k,SU}(r w_k, N_0)} \le 1 \right\}. \tag{13}$$

This measure captures the loss in bit-error rate (BER) performance of a detector in the high SNR region where the dominant degradation in performance is due to the presence of interfering users rather than additive noise. In particular, a detector with a non-zero value of asymptotic efficiency $\eta_k(\phi)$ for user k operating in high SNR conditions has a BER performance that rivals the BER of the single-user RAKE receiver designed for the k^{th} single user FSRF channel with average received SNR per bit that is equal to $w_k \eta_k(\phi)$.

We will first state the result for the asymptotic efficiency of users in the first group because the first group detector ϕ_{G_1} is conceptually simpler than the subsequent detectors $\tilde{\phi}_{G_j}$. It does not involve the interference cancellation step. For convenience, let us consider the group variable $G = \{1, \cdots, |G|\}$. Let $\mathbf{F}_k(G)$ be the correlation matrix of the signal vector obtained by stacking $\mathbf{u}_k(t)$ and the interfering signal vector $\mathbf{z}(t)$ which in turn is defined as being the result of stacking signal vectors of users $|G| + 1$ to K. Writing this correlation matrix as a 2×2 block partitioned matrix, we have

$$\mathbf{F}_k(G) \triangleq \int_{(L_M-1)/W}^{T} \begin{pmatrix} \mathbf{u}_k(t) \\ \mathbf{z}(t) \end{pmatrix}^* \begin{pmatrix} \mathbf{u}_k(t) \\ \mathbf{z}(t) \end{pmatrix}^T dt = \begin{bmatrix} \mathbf{X}_k(G) & \mathbf{Y}_k^{*T}(G) \\ \mathbf{Y}_k(G) & \mathbf{Z}_k(G) \end{bmatrix}, \tag{14}$$

with $\mathbf{z}(t) \triangleq [\mathbf{u}_{|G|+1}^T(t), \cdots, \mathbf{u}_K^T(t)]^T$ is an interfering signal vector of length $\sum_{l=|G|+1}^{K} L_l$.

Proposition 2 *The asymptotic efficiency of the first group detector ϕ_{G_1} for the k^{th} user in G_1 admits the formula*

$$\eta_k(\phi_{G_1}) = \left\langle \left\{ \lambda_i \left(\mathbf{I} - [\mathbf{X}_k(G_1)]^{-1} \mathbf{Y}_k^{*T}(G_1) [\mathbf{Z}_k(G_1)]^{-1} \mathbf{Y}_k(G_1) \right) \right\}_{i=1}^{L_k} \right\rangle_{\mathcal{G}}, \tag{15}$$

where the matrices involved are defined according to (14). Also, $\lambda_i(\mathbf{A})$ denotes the i^{th} eigenvalue of the matrix $\mathbf{A}$ and $\langle \{a_i\}_{i=1}^n \rangle_{\mathcal{G}}$ denotes the geometric mean of the numbers $\{a_i\}_{i=1}^n$ defined as $(\prod_{i=1}^n a_i)^{1/n}$.

The above proposition was proved in [3]. It was shown in that paper that for $k \in G \subset H$, $\eta_k(\phi_G) \leq \eta_k(\phi_H)$, i.e., increasing the group size results in an improved performance. An insight into this result can be gleaned from the following argument. When $k = 1$ and $|G| = 1$, the detector $\phi_{\{1\}}$ is a decorrelator-combiner. Its performance can be significantly poorer than the optimum detector [3]. Now suppose we consider the case where k is still 1, but $|G|$ is greater than 1. In this case, note that by the manner in which the definition of $\mathbf{z}(t)$ changes, the above formula shows that the performance of the group detector ϕ_G is equivalent to that of a decorrelating combiner for user 1 when considered over the fictitious channel where the rest of the users within the first group are absent. This explains not only why increasing the group size results in an improvement in performance, but also gives an insight about the mechanism through which the improvement results.

Next, let us define super-groups $S_m = \cup_{l=1}^m G_l$ for $1 \leq m \leq J$. Consider the super-group detector ϕ_{S_m} which is the (non-sequential) group detector for users in the super-group S_m. Let

p denote the i^{th} user in group G_m (i.e., $p = i + K_{m-1}$). The performance of this super-group detector bounds from above the performance of the sequential group detector $\tilde{\phi}_{G_m}$ for each user $i \in G_m$. This is the result of the following proposition.

Proposition 3 *The asymptotic efficiency of the m^{th} sequential group detector $\tilde{\phi}_{G_m}$ for the i^{th} user of group G_m (i.e., user $p = i + K_{m-1}$) is upper bounded by the asymptotic efficiency of the super-group detector ϕ_{S_m} for that user, which in turn admits the formula given below:*

$$\eta_p(\tilde{\phi}_{G_m}) \leq \eta_p(\phi_{S_m}) = \left\langle \left\{ \lambda_i \left(\mathbf{I}_{L_p} - [\mathbf{X}_p(S_m)]^{-1} \mathbf{Y}_p^{*T}(S_m) [\mathbf{Z}_p(S_m)]^{-1} \mathbf{Y}_p(S_m) \right) \right\}_{i=1}^{L_p} \right\rangle_{\mathcal{G}}, \qquad (16)$$

where the matrices involved are defined according to (14) with the appropriate change of variables (k replaced by p and the group variable G replaced by the super-group S_m).

We have the interpretation that the super-group detector behaves as a decorrelating-combiner for each user in group G_m when evaluated in a fictitious channel where only the user in question and the interfering users outside the super-group, namely, the interfering users from future group $G_{m+1}, \cdots, G_J$ are active, and the rest of the users within the super-group are inactive. The super-group detector asymptotic efficiency bound therefore also coincides with the perfect interference cancellation bound. This is unlike the result for the Gaussian multiuser channel [5]. In any case, as the group index m increases, the super-group S_m becomes larger and therefore, the upper bound tends to increase with m. This is as it should be, because the interference is being cancelled from a larger number of users. Loosely speaking therefore, a good strategy for ordering the users is in accordance with decreasing signal powers so that the effects of error propagation are minimal and the super-group or perfect cancellation upper bound is tight. In this case the decreasing signal powers are compensated by the increasing asymptotic efficiencies with the result that the effective average signal to noise ratios are acceptable for all users. On the other hand, the upper bound in Proposition 3 can be overly optimistic if the effective average signal to noise ratios of the users in the previous groups were to be less than the effective average signal to noise ratios expected from the bound for the users in the current group.

The next proposition specifies a lower bound on the asymptotic efficiency and thereby includes the deleterious effects of error propagation that can result from interference enhancement rather than interference cancellation due to errors in previous groups.

Proposition 4 *The asymptotic efficiency of the sequential group detector $\tilde{\phi}_{G_j}$ for the k^{th} user of group G_j (or the q^{th} user where $q = k + K_{j-1}$) is lower bounded as*

$$\eta_q(\tilde{\phi}_{G_j}) \geq \min \left\{ \min_{1 \leq m \leq j-1} \; \min_{p \in G_m;\, s.t.\, L_p \leq L_q} \gamma_p \frac{w_p}{w_q} \eta_p(\phi_{S_m}) \, , \, \eta_q(\phi_{S_j}) \right\} \qquad (17)$$

where $\eta_p(\phi_{S_m})$ is specified in, and $\eta_q(\phi_{S_j})$ can be deduced from, Proposition 3. The function γ_p is defined as

$$\gamma_p = \begin{cases} 0 & \text{if } L_p < L_q; \\ \left\langle \left\{ \lambda_l (\mathbf{\Sigma}_{pp} \mathbf{\Sigma}_{qq}^{-1} \mathbf{R}_{pp} \mathbf{R}_{qq}^{-1}) \right\}_{l=1}^{L_q} \right\rangle_{\mathcal{G}} & \text{if } L_p = L_q \; . \end{cases} \qquad (18)$$

Consequently, when the number of paths of all users are equal, a set of sufficient conditions for the sequential group detector $\tilde{\phi}_{G_j}$ to achieve the super-group detector bound for user $q \in G_j$ are as follows:

$$\eta_q(\tilde{\phi}_{G_j}) = \eta_q(\phi_{S_j}) \quad \text{if} \quad \frac{w_p}{w_q} \geq \frac{\eta_q(\phi_{S_j})}{\gamma_p \eta_p(\phi_{S_m})} \qquad \forall 1 \leq p \leq K_{j-1} \tag{19}$$

The above proposition states the conditions under which error propagation effects disappear as SNR increases and quantifies the ad-hoc notion that users must be arranged in the order of decreasing signal strengths. Note that these conditions depend on the channel state statistics, the signal correlations and the total average received signal powers, so that they can be easily monitored. They suggest efficient means for ordering the users so as to minimize the effects of error propagation. To be sure, the sufficient conditions indicate that if the total average received signal powers are sufficiently disparate, then the ordering of users according to decreasing total average received signal powers suffices. This may not always be the case. Furthermore, for a given set of signature waveforms, average SNRs, and channel statistics, if error propagation problems are unavoidable with unity group sizes for any ordering of the users, the problem can be alleviated to a large extent by considering larger group sizes.

5 Numerical Examples

Consider a six user direct-sequence spread-spectrum system employing Gold sequences of length 31 of [3] operating in a fading multipath environment with four paths for each user. The users are numbered according to decreasing average power ratios (with respect to the minimum power) given in order as $[10.0, 2.5, 2.0, 1.5, 1.25, 1.0]$. Suppose that the performance of a single-user RAKE receiver for the last (weakest) user in the hypothetical single-user scenario, where all the other users are absent, is considered acceptable. Equivalently, the effective SNR (ESNR) to minimum actual SNR ratio (henceforth referred to relative ESNR) for every user has to be no less than 1. The decorrelating combiner results in relative ESNRs for the six users given in order as $[1.40, 0.37, 0.53, 0.21, 0.18, 0.26]$. As for sequential group detection, it turns out that the decorrelating-combiner based decision-feedback detector suffices. The resulting upper bounds for the relative ESNRs for the six users are $[1.40, 1.22, 1.12, 1.06, 1.09, 1.0]$ and the lower bounds are $[1.40, 1.22, 1.12, 1.06, 1.06, 1.0]$. The minimum specification is met. Moreover, note that the upper and lower bounds coincide for all but the fifth user. Next, suppose that the average received signal to noise ratios (relative to the weakest user) for the six users are $[2.5, 2.5, 1.5, 1.5, 1.0, 1.0]$. The relative effective average signal to noise ratios (relative to the actual SNR of weakest user) for the six users for the suboptimal (non-sequential) decorrelating-combiner are $[0.35, 0.37, 0.39, 0.21, 0.15, 0.26]$. So this scheme is inadequate. The upper bounds for the relative ESNRs for the six users for the decorrelating combiner based decision feedback detector with users arranged in order of decreasing SNRs are $[0.35, 1.22, 0.84, 1.06, 0.87, 1.0]$ with a lower bound of 0.35 for all the users. While some of the pessimism of the lower bounds may be attributed to its looseness, the rest of it has to do with the deleterious effects of error propagation from imperfect cancellation. Note that this is the case inspite of the users being arranged in decreasing order of signal strengths. Next, consider a sequential group detector with an ordered group partition $\{1,2\}\{3,4\}\{5,6\}$ consisting of three groups of size two each. The upper and lower bounds coincide for every user and the relative ESNRs for the six users are equal to $[1.15, 1.21, 1.0, 1.06, 1.0, 1.0]$. The minimum requirement is thus met. Error propagation has little effect on the performance. The lower complexity of the sequential group detector however is achieved at the expense of no more than a 3.4 dB loss for the strongest users and less than a 1.8 dB loss for the users of intermediate power relative to single-user performance in a single-user channel.

References

[1] J. G. Proakis, "*Digital Communications*," Third Ed., Mc-Graw Hill, NY, 1995.

[2] M. Schwartz, W. R. Bennett and S. Stein, *Communication Systems and Techniques*," Inter-University Electronics Series, Vol-4, McGraw-Hill Book Company, 1966.

[3] M. K. Varanasi, "Group Detection for Synchronous CDMA Communication Over Frequency-Selective Fading Channels," *Proc 31st Annual Allerton Conf on Communication, Control, and Computing*, Sep 29-Oct 1, 1993, pp. 849-858.

[4] A. Duel-Hallen, "Decorrelating Decision Feedback Multiuser Detector for Synchronous Code-Division Multiple-Access Channel," *IEEE Trans Commun* **COM-41**:2, 285-290, February 1993.

[5] M. K. Varanasi, "Group Detection for Synchronous Gaussian Code-Division Multiple-Access Channels," *IEEE Trans Inform. Th.*, **41**:4 July 1995.

[6] M. K. Varanasi, "Parallel Group Detection for Synchronous Synchronous CDMA Communication Over Frequency-Selective Rayleigh Fading Channels," *IEEE Trans Inform. Th.*, **43**:1 (January 1996).

A New Coding Method for the Asynchronous Multiple Access Channel

Alex J. Grant and Lars K. Rasmussen
Mobile Communications Research Centre
University of South Australia
Warrendi Rd. Pooraka 5095 Australia

Abstract

A new coding method, based on unequal error protection and successive cancellation is proposed for the asynchronous multiple access channel. It is shown that this method can achieve any point in the capacity region. The number of decoding steps required is the same as that for a successive cancellation scheme operating at a vertex of the corresponding synchronous channel.

1 Introduction

The design of good codes for the multiple access channel has proved to be a difficult problem. One must find a set of codebooks which, in the absence of noise, may be uniquely decoded at the receiver. These difficulties are compounded upon the addition of noise, or the loss of block synchronism.

Consider a general block asynchronous, symbol synchronous m-user multiple access channel, as depicted in Figure 1. The channel is defined by m sources, transmitting information to a common receiver. User i transmits messages x_i at rate R_i, from a source alphabet $\mathcal{X}_i$, according to a distribution $p_i(x_i)$. The received signal is Y, taking values from $\mathcal{Y}$ according to the transition probabilities $q(y|x_1, x_2, \ldots, x_m)$. The region of rates in which transmission is possible with arbitrarily low error probability, for a particular set of source distributions is well understood [1] and is given by the set of rates $\underline{R} = (R_1, R_2, \ldots, R_m)$ satisfying, for all $\mathcal{I} \subseteq \{1, \ldots, m\}$ the following constraint

$$\sum_{i\in\mathcal{I}} R_i \leq I\left((X_i)_{i\in\mathcal{I}}; Y \mid (X_i)_{i\notin\mathcal{I}}\right). \tag{1}$$

The capacity region is the union of such regions, over the family of product distributions $\prod_{i=1}^{m} p_i(x_i)$. It is known that the vertices of the capacity region can be achieved by successive interference cancellation [1]. The users are decoded in a specific order, for

---

This work was supported in part by Telstra under Contract No.7368 and by the Commonwealth of Australia under International S & T Grant No.56.

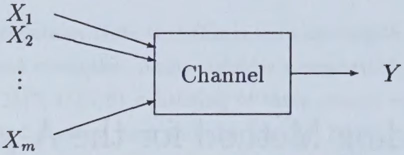


Figure 1: m-user multiple access channel.

example according to their index number, i. Each user transmits using a code sufficiently powerful to ensure error-free transmission, treating users with higher indices as noise. At such points in the capacity region, we only need to design the appropriate single user codes.

It is a consequence of Caratheodory's theorem [2] that any rate point in the convex hull of such vertices can be achieved by "time sharing" between the vertices. Time sharing unfortunately, introduces the need for strict block synchronisation between users (neccessitating a global clock) which imposes severe constraints on the system. In fact, if such synchronism is available, the capacity region can be expanded to include any extra points yielded by the convex hull of the region described by (1).

Recent advances [3, 5] have shown however, that it is possible to transform the multiple access channel problem for any point in the capacity region into series of single user coding problems. In particular it has been shown that the problem of coding for the m user discrete memoryless multiple access channel can be solved through the use of at most $2m - 1$ decoding steps, instead of the m steps required if operating at a vertex. In these schemes, all users but one is split into two pseudo-users. Successive interference cancellation is then performed for this expanded system, treating all other pseudo-users, including the pseudo-user corresponding to the user of interest, as noise. By appropriately selecting rates and distributions for the pseudo-users, a corner point in the capacity region for $2m-1$ users, corresponding to the required rate point is achieved. An important fact to note is that the input distributions of the two pseudo-users corresponding to a specific user are not necessarily the same as the input distribution of the original user.

We propose a new coding method which has lower complexity than the schemes in [3, 5], which also renders certain additional advantages in terms of the design of the the single user codes required.

Unequal error protection of the information, in conjunction with successive cancellation decoding is used. The source data is split into two streams which are encoded with unequal rate codes. The resulting two codeword streams are then randomly interlaced to form a single stream. Codeword alignment is not required, allowing block asynchronism. Also, each code uses the same input alphabet and input distribution as originally required to achieve a vertex. Specifically, if equi-probable input symbols are used, traditional single user code design techniques are applicable. In contrast to the techniques described in [3, 5], the proposed method only requires the same decoding effort as a decoder operating at a vertex, i.e. m decoding steps instead of $2m - 1$. We will further show that by appropriatly interpreting the interlacing procedure, this new scheme can be viewed as a convenient special case of the user splitting approach in [3].

The paper is organised as follows. In Section 2 the code construction is described. The capacity of the proposed system is analysed in Section 3, where it is shown that the scheme can achieve any rate point in the capacity region. Concluding remarks are made in Section 4.

2 Code Construction

In this section we shall describe the encoding and decoding procedure for the proposed scheme.

2.1 Encoding

With reference to Figure 2, let us consider a particular user, X. The information stream of X shall be encoded as follows. Unequal error protection shall be used, such that codewords of length n, consist of λn coded bits at rate R_1, and the remaining $(1-\lambda)n$ coded bits at rate R_2, where $0 \leq \lambda \leq 1$. We shall refer to λ as the *self time sharing parameter* for user X. The labels on the paths in Figure 2 denote the size of a frame at each point in the encoder. We thus divide the input stream into frames of length $\lambda n R_1$, which are fed to the rate R_1 encoder, and frames of length $(1-\lambda)nR_2$, which are fed to the rate R_2 encoder. The two resulting codeword frames, of length λn and $(1-\lambda)n$ are combined to form the final n bit frame by a process of interlacing, which works as follows. For each new frame, the interlacer selects a vector $\underline{l}$ of length λn whose elements are distinct random integers, l_i from the range $[0, n-1]$. The codeword frame is then formed by placing the bits from the first encoder consecutively at positions $l_0, l_1, \ldots, l_{\lambda n-1}$, and filling the remaining positions with the bits from the other encoder. Figure 3 depicts this process for $n = 13$, $\lambda = \frac{5}{13}$ and the interlacing parameter $\underline{l} = (3, 5, 6, 10, 11)$.

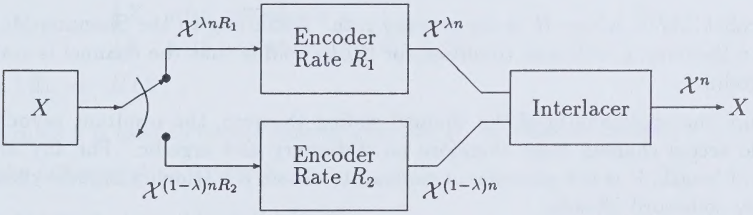

$\mathcal{X}^{\lambda n R_1}$
Encoder Rate R_1
$\mathcal{X}^{\lambda n}$
X
Interlacer
$\mathcal{X}^n$
X
Encoder Rate R_2
$\mathcal{X}^{(1-\lambda)nR_2}$
$\mathcal{X}^{(1-\lambda)n}$

Figure 2: Coding Scheme.

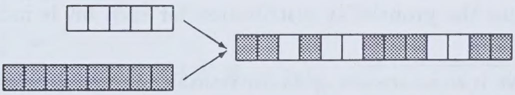


Figure 3: Interlacing process.

2.2 Decoding

The decoding of the m users shall be performed by successive cancellation, according to some pre-arranged order of decoding, treating the rate R_1 and rate R_2 bits from each user as individual "pseudo" users, which we shall denote U and V respectively. For example, if the first pseudo-user in the decoding order list is U_1, then the decoding procedure is as follows. The U_1 bits are extracted from Y (the interlacer parameters are known to the decoder) and decoded. The corresponding (possibly modified) data is then subtracted from the appropriate positions in Y, creating a new output stream, Y' [1]. This process is continued for the remaining pseudo users, each stage taking the Y' output from the previous stage as it's input, Y.

If the rates of the pseudo-users are chosen such that they can be decoded with arbitrarily low error probability, while treating users further down the decoding order as noise, then the entire system can be made to operate with vanishing error probability.

At this point, it may seem that the interlacing operation is superfluous, but it is desirable for two reasons. Since we are using successive cancellation, the absence of the interlacing process would cause the channel to appear bursty, since undecoded pseudo-users would appear in blocks. The introduction of the interlacers avoids the resulting requirement to design block-error correcting codes.

In the following sections, we analyse the coding scheme using an information theoretic approach, which leads us to the second reason. The channel coding theorem relies on the properties of jointly typical sequences [1], which in turn rely on the weak law of large numbers, specifically that for a stochastic process $\{X_n\}$

$$-\frac{1}{n}\log p(X_0, \ldots, X_{n-1}) \to H\,, \tag{2}$$

with probability 1, where H is the entropy rate. According to the Shannon-McMillan-Breiman theorem, a sufficient condition for (2) to hold is that the channel is stationary and ergodic.

To ensure the applicability of the channel coding theorem, the resultant asynchronous multiple access channel must therefore be stationary and ergodic. For any arbitrary codeword length, it is not possible, in general, to obtain a stationary ergodic channel for arbitrary codeword off-sets.

A stationary ergodic channel can however be obtained by introducing a random interlacer for each user. It follows that the probability distribution for each bit in a codeword is now identical, i.e., the probability for a bit in a codeword being encoded at a given rate is the same for all bits. It is then clear that the channel is stationary and ergodic for any codeword off-set since the probability distribution for each bit is independent of other users.

---

[1] In cases where there is no subtraction operation available, the next decoder will perform typical sequence decoding, conditional on the already decoded data.

3 Capacity Analysis

We now turn our attention to analysing the capacity of the proposed scheme. In particular we shall see that the scheme can achieve any point in the capacity region.

3.1 Two User Channel

For the two-user channel we have the following theorem

Theorem 1 (Two-User Case) *Any rate point in the capacity region of a two-user asynchronous multiple access channel may be achieved by the proposed scheme.*

For the proof of this theorem we shall need the following two lemmas.

Lemma 1 *Let U, V completely determine X, then*

$$I(X, W; Y) = I(U, V, W; Y)$$

Proof: Using the fact that $I(A; B) = I(B; A)$ and applying the data processing inequality [1], to the Markov chain $U, V, W \to X, W \to Y$, we have

$$I(X, W; Y) \geq I(U, V, W; Y) ,$$

with equality if and only if $I(X, W; Y \mid U, V, W) = 0$, which is true since by assumption, U, V completely determine X and W is completely determined by itself. ■

Lemma 2 *Let X, W and Z be mutually independent random variables, and let Z_λ be defined as follows*

$$\begin{aligned} Z_\lambda &= \begin{cases} Z & \text{with probability } \lambda, \\ \emptyset & \text{with probability } 1-\lambda \end{cases} \\ H(Z \mid \emptyset) &= H(Z) , \end{aligned}$$

(The symbol $\emptyset$ is used to denote that in this case Z_λ provides no information about Z.) Then the following equality holds

$$\begin{aligned} I(X; Y \mid W, Z_\lambda) &= \lambda I(X; Y \mid W, Z) \\ &\quad + (1-\lambda) I(X; Y \mid W) . \end{aligned}$$

Proof: From the definition of conditional mutual information and the independence of X, W and Z_λ it is true that

$$\begin{aligned} I(X; Y \mid W, Z_\lambda) &= \\ &\sum_{z_\lambda} p(Z_\lambda = z_\lambda) I(X; Y \mid W, Z_\lambda = z_\lambda) \\ &= \lambda \sum_{z} p(Z = z) I(X; Y \mid W, Z = z) \\ &\quad + (1-\lambda) I(X; Y \mid W, Z_\lambda = \emptyset) \\ &= \lambda I(X; Y \mid W, Z) + (1-\lambda) I(X; Y \mid W) . \end{aligned}$$

Which was what was to be proved. ■

Corollary 1 *It follows that $I(X;Y \mid W, Z_\lambda)$ is a continuous function of λ, and is bounded,*

$$I(X;Y \mid W) \leq I(X;Y \mid W, Z_\lambda) \leq I(X;Y \mid W, Z) .$$

Corollary 2 *Furthermore, if the distributions of the random variables, X, W and Z do not depend on λ then $I(X;Y \mid W, Z_\lambda)$ is in fact a linear function of λ.*

We are now placed to provide a proof of Theorem 1.

Proof: We shall restrict our attention to achieving the overall constraint,

$$R_1 + R_2 = I(X_1, X_2; Y) , \tag{3}$$

and ensure that we are not already on a corner point by imposing the strict inequalities

$$\begin{array}{ccccc} I(X_1;Y) & < & R_1 & < & I(X_1;Y \mid X_2) \\ I(X_2;Y) & < & R_2 & < & I(X_2;Y \mid X_1) . \end{array}$$

Let X_1 be encoded using the proposed method. The associated two pseudo-users, U and V completely determine X_1. Hence by Lemma 1 and (3),

$$R_1 + R_2 = I(U, V, X_2; Y) .$$

Which, by the chain rule for mutual informations may be written as

$$\begin{aligned} I(U, V, X_2; Y) &= I(U;Y) + I(X_2;Y \mid U) \\ &\quad + I(V;Y \mid X_2, U) . \end{aligned}$$

At this point note that this is in a form suitable to be achieved by successive cancellation. All that is required is that the corresponding individual rares, R_1 and R_2 may be achieved. By Lemma 2 we know that

$$I(X_2;Y) < I(X_2;Y \mid A) < I(X_2;Y \mid X_1)$$

Hence it is possible to set $R_2 = I(X_2;Y \mid A)$, with appropriate choice of λ. It follows that $R_1 = I(A;Y) + I(B;Y \mid X_2, A)$. Thus the point (R_1, R_2) may be achieved by the proposed scheme. ■

3.2 m User Channel

Having proved the case for two users, the generalisation to 3 or more user channels follows directly from the induction presented in [3]. This is not completely fortuitous, since as we shall see in the next section, this scheme is related to the one therin described.

One point of note however, is that the inductive proof only gives that at most $2m - 1$ pseudo-users are required. It does not imply that each user is associated with only two pseudo-users. Hence it is possible that the algorithm mirroring the inductive proof will result in unequal error protection involving more than two encoders for some users, and only one encoder for others.

We do however conjecture the following

Conjecture 1 It is possible to achieve any rate point in the capacity region of the m user asynchronous multiple access channel with the proposed scheme, where each user employs not more than two pseudo-users.

For the Gaussian multiple access channel system of [5] this conjecture has been given proof in [4].

4 Discussion

We have presented a new coding scheme for the m user asynchronous multiple access channel, and have shown that using this scheme, any point in the capacity region may be obtained. This scheme bears many similarities to the system described in [3], and some of the analysis is somewhat parallel.

4.1 Relationship to earlier systems

The new coding technique is closely related to the system of [3], where we replace the combining function, $\max\{.,.\}$ with the interlacer, i.e. instead of having a function $X = f(U, V)$, we generalise to allow $X = f(U, V, t)$ where t is the time variable. In particular,

$$f(U, V, t) = \begin{cases} U^+ & (t \mod n) \in \underline{l} \\ V^+ & \text{otherwise} \end{cases}$$

where the superscript U^+ denotes the next symbol in the U stream etc.

The new scheme however displays some important advantages, which increase the practicality of such "Rate Division Multiple Access" systems.

4.2 Advantages

We now summarise the advantages of this new system, paying particular attention to pratical issues of code design, and contrast to the systems mentioned in [3, 5].

The problem of coding for the m user asynchronous multiple access channel has been transformed into $2m - 1$ single user coding problems. Thus the problem of designing codes specifically for multiple access channels, an area which has proved its difficulty, has been avoided. All the theory developed for single user channels can be applied directly to the multiple access channel.

Since both pseudo-users associated with a particular user have codebooks drawn from the same alphabet and with the same distribution as the original user, we avoid the neccessity of re-designing the codebooks if the decoding order changes, as would be required by the systems described in [3]. Furthermore, through use of expurgating or puncturing, each user's pseudo-users can use subsets of one "generic" codebook, which contains enough codewords to allow trasmission at the highest expected rate.

At the decoder, we only require the same number of decoding steps as required for a synchronous system operating at a vertex, which is due to the interlacing operation used to merge the two pseudo-users. For example, for the synchronous system to decode m users, each having an L bit frame, mL bits must be decoded, which is exactly the same for the proposed system. This may be contrasted to [3, 5], where $(2m-1)L$ bits must be decoded. In addition, a pseudo-user does not consider the other pseudo-user making up the whole suer as noise, which would reduce the complexity required for each pseudo-user's code.

An additional advantage that once the decoding order for the system has been decided, the λ values for each user may be simply found as the solution to a set of linear equations.

Acknowlegements

The authors would like to thank Phil Whiting for many helpful discussions.

References

[1] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley, 1991.

[2] Eggleston H. G. *Convexity.* Cambridge tracts in mathematics and mathematical physics. Cambridge University Press, London, 1958.

[3] A. Grant, B. Rimoldi, R. Urbanke, and P. Whiting. Single user coding for the discrete memoryless multiple access channels. In *IEEE Int. Symp. Info. Theory*, 1995.

[4] B. Rimoldi. RDMA/IC on AWGN with at most two pseudousers per user. Personal Communication, July 1995.

[5] B. Rimoldi and R. Urbanke. On single-user decodeable codes for the Gaussian multiple access channel. In *IEEE Int. Symp. Info. Theory*, 1994.

A Linear Receiver for the Coded Asynchronous Multiuser CDMA Channel

Paul D. Alexander & Lars K. Rasmussen
Mobile Communications Research Centre
Signal Processing Research Institute
University of South Australia
Warrendi Road, The Levels
SA 5095, Australia
E-Mail: palex@spri.levels.unisa.edu.au
FAX: +61 8 302 3873

Christian Schlegel
Coding, Communications,
and Control (C^3) Lab.
The University of Texas at San Antonio
San Antonio
TX 78249, USA
E-Mail: chris@beast.eng.utsa.edu
FAX: +1 210 691 5589

Abstract

In this paper we consider a multiuser CDMA system with error control coding. The multiple access interference and the error control decoding are handled separately at the receiver. The error control decoding is implemented via single user soft–input decoders utilising metrics from multiuser metric generators. These metric generators are linear complexity multiuser receivers where the resulting output decision statistic is interpreted as a metric. The decorrelator, and a new scheme termed the projection receiver are utilised as metric generators. The coded performance of the projection receiver metric is shown to be superior to the decorrelator.

1 Introduction

The majority of research on receivers for multiuser CDMA communications systems has been in the area of uncoded receiver design [1, 2, 3, 4, 5]. Once the multiuser interference (MUI) has been resolved, error control coding can be applied independently. In this case hard decisions on the coded bits of each user are made prior to decoding. Clearly, this segmented design philosophy is suboptimal since we are throwing away information by making hard decisions. Conversely, the system that jointly considers MUI and error control coding is prohibitively complex. Information theory tells us that in order to get close to the capacity we should invest in the design of a joint error control codebook for the users rather than a joint spreading codebook [6]. However, the design of joint error control codes is also prohibitively complex.

The simplicity of spreading code design makes a traditional CDMA system attractive, especially for an asynchronous channel. The receiver structures considered here are intended for the uplink in a mobile communications system. Time variations of the channel translate into considerable network complexity when a synchronous channel CDMA is maintained. The asynchronous CDMA channel is therefore studied in this work since it

---

This work was supported in part by Telstra Australia under Contract No.7368 and by the Commonwealth of Australia under International S & T Grant No.56.

offers a realistic trade–off between performance, receiver complexity and network complexity.

In this paper we present a new type of metric generator. This device is derived based on geometrical properties and it is thus termed the projection receiver (PR). The initial derivation for the PR for the uncoded synchronous channel can be found in [7]. We compare the coded performance of the PR and decorrelator (DC) [2, 8] fed decoders and show that the PR fed system performs at least as well as the DC fed system.

The paper is organised as follows. In Section 2 we introduce the uplink CDMA communication system where error control coding is employed. A new metric generator is introduced in Section 3. The uncoded and coded performance of the system under different metric sources are analysed in Section 4, and simulation results are presented in Section 5. Throughout this paper scalars are lower case, vectors are underlined lower case, and matrices are underlined upper case. Subscripting is dropped where no ambiguities arise. The symbols $(\cdot)^{\mathsf{T}}, (\cdot)^{-1}$ and $\|\cdot\|$ are the transposition, inversion and vector norm operators respectively. The delimiter $\{\cdot\}^y$ defines a space of dimension y, and the $\lfloor x \rfloor$ operator returns the largest integer not greater than x.

2 Uplink Model for Symbol Asynchronous CDMA

In this section, the uplink model for the CDMA communication system considered throughout this paper shall be briefly described. The uplink model is based on a very simple discrete–time symbol asynchronous, chip synchronous CDMA system assuming single path channels and the presence of stationary additive white Gaussian noise with zero mean and variance $\sigma^2 = N_0/2$. For simplicity the system model considered in this paper does not account for ISI. However, the effect of MUI which is basically the limiting factor in a CDMA communication system is explicitly included.

The K users in this communication system first encode their information sequence $\underline{b}_k \in \{-1,1\}^{M_u}$ via a (n,k) encoder into a sequence $\underline{d}_k \in \{-1,1\}^M$ where $M = (n/k)M_u$. They then spread this encoded sequence by choosing a spreading code $\underline{s}_{k,m} \in \{-1,1\}^L$ which is multiplied by the coded bit $d_{k,m}$, $m \in [0, M)$, and transmitted using BPSK over the AWGN channel. The spreading codes transmitted by each user in any given symbol (codeword bit) interval are assumed to arrive at the receiver antennae in a symbol asynchronous and chip synchronous manner. The asynchronous attribute of the uplink is described by assigning a fixed delay τ_k to each user. τ_k is defined as the delay of the arrival of the users spread waveform at the receiver with respect to some external symbol clock. We assume for simplicity that these delays are integer multiples of chip durations which allows us to express the output of a chip waveform matched filter as a linear combination of shifted spreading codes, where the shifts are determined by τ_k. Specifically,

$$\underline{e} = \underline{A}\underline{d} + \underline{n},$$

where

$$
\begin{aligned}
\underline{e} &\in \mathbb{R}^{(M+1)L} \\
\underline{A} &= (\underline{a}_0, \cdots, \underline{a}_{MK-1}) \in \{-1,0,+1\}^{(M+1)L,MK} \\
\underline{a}_j &= \left(\underline{0}_{\eta(j)L+\tau_{\kappa(j)}}, \underline{s}_{\kappa(j),\eta(j)}, 0, \cdots, 0\right)^{\mathsf{T}} \in \{-1,0,+1\}^{(M+1)L} \\
\kappa(j) &= (j \bmod K) + 1 \\
\eta(j) &= \lfloor j/K \rfloor \\
j(k,m) &= mK + k - 1 \\
\tau_k &\in \{0, 1, \cdots, L-1\} \\
\underline{d} &= (d_0, \cdots, d_{MK-1})^{\mathsf{T}} \in \{-1,1\}^{MK} \\
\underline{n} &\sim \left(N(0,\sigma^2), \cdots, N(0,\sigma^2)\right)^{\mathsf{T}} \in \mathbb{R}^{(M+1)L}
\end{aligned}
$$

At the input to this channel are K encoders and at the output is the multiuser receiver that generates metrics for the K soft input single user decoders. The decoders form their branch metric for a particular codeword by taking n metrics from the metric streams feeding the decoder. For example, if `101` was a codeword in user k's codebook then the corresponding branch metric would be $\lambda_{k,m}(+1) + \lambda_{k,m+1}(-1) + \lambda_{k,m+2}(+1)$. The discrete–time communications system model is shown in Fig. 1 where the concept of the metric streams becomes apparent.

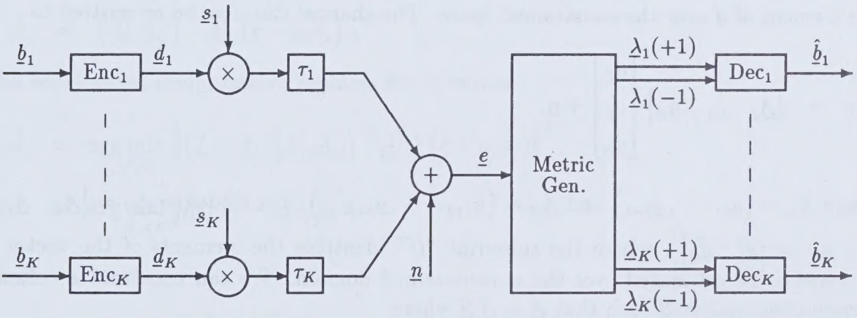

$\underline{b}_1$
Enc$_1$
$\underline{d}_1$
$\underline{s}_1$
$\times$
τ_1
$\underline{b}_K$
Enc$_K$
$\underline{d}_K$
$\underline{s}_K$
$\times$
τ_K
$+$
$\underline{e}$
$\underline{n}$
Metric Gen.
$\underline{\lambda}_1(+1)$
$\underline{\lambda}_1(-1)$
$\underline{\lambda}_K(+1)$
$\underline{\lambda}_K(-1)$
Dec$_1$
$\hat{\underline{b}}_1$
Dec$_K$
$\hat{\underline{b}}_K$

Figure 1: Communications System for Uplink.

3 Linear Metric Generation

We now describe two linear methods for generating the metrics for the decoders. Each stream consists of the Euclidean distance from the continuous estimate made by the multiuser receiver to each of the two possible hypotheses, namely -1, and +1.

3.1 Projection Receiver

With sequence estimation we face the problem of detecting d_j, $j \in [0, MK)$ over the asynchronous CDMA channel

$$\underline{e} \;=\; \underline{A}\underline{d} + \underline{n}.$$

The constrained maximum likelihood sequence estimate of $\underline{d}$ is

$$\hat{\underline{d}} \;=\; \arg \min_{d \in \mathcal{D}^{MK}} \|\underline{e} - \underline{A}\underline{d}\|^2 ,$$

where $\mathcal{D} \equiv \{-1, +1\}$. The complexity of such a constrained search is prohibitive. Let us pick a particular transmission, say m, by a particular user, say k, to detect and partition the transmitted (codeword bits) $\underline{d}$ as

$$\underline{d} \;=\; \begin{bmatrix} \underline{d}_L^{\mathsf{T}} & d_j & \underline{d}_R^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}} ,$$

where $j = \jmath(k, m)$, and

$$\begin{aligned} \underline{d}_L &= (d_0, \cdots, d_{j-1})^{\mathsf{T}} , \\ \underline{d}_R &= (d_{j+1}, \cdots, d_{MK-1})^{\mathsf{T}} . \end{aligned}$$

The main property of the projection receiver is to save complexity by estimating $\underline{d}_L$ and $\underline{d}_R$ over the real domain, i.e., $\hat{\underline{d}}_L \in \mathbb{R}^j$ and $\hat{\underline{d}}_L \in \mathbb{R}^{MK-j-1}$, and only d_j over the constrained domain, ie. $d_j \in \mathcal{D}$. Note that we could have chosen to estimate more than one element of $\underline{d}$ over the constrained space. The channel can now be re-written as

$$\underline{e} \;=\; \begin{bmatrix} \underline{A}_L & \underline{a}_j & \underline{A}_R \end{bmatrix} \begin{bmatrix} \underline{d}_L \\ d_j \\ \underline{d}_R \end{bmatrix} + \underline{n}.$$

where $\underline{A}_L = \left(\underline{a}_0, \cdots, \underline{a}_{j-1}\right)$ and $\underline{A}_R = \left(\underline{a}_{j+1}, \cdots, \underline{a}_{MK-1}\right)$. Let us define $\underline{A}_U = \begin{bmatrix} \underline{A}_L & \underline{A}_R \end{bmatrix}$, and $\underline{d}_U = \begin{bmatrix} \underline{d}_L^{\mathsf{T}} & \underline{d}_R^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}$, where the subscript "$U$" identifies the elements of the vector $\underline{d}$ that are to be estimated over the unconstrained domain. We can construct a column permutation matrix $\underline{S}$ such that $\underline{A} = \underline{A}'\underline{S}$ where

$$\underline{A}' \;=\; \begin{bmatrix} \underline{A}_L & \underline{A}_R & \underline{a}_j \end{bmatrix} = \begin{bmatrix} \underline{A}_U & \underline{a}_j \end{bmatrix} .$$

Clearly, $\underline{A}'$ is a column-wise permutation of $\underline{A} = \begin{bmatrix} \underline{A}_L & \underline{a}_j & \underline{A}_R \end{bmatrix}$. Similarly $\underline{d} = \underline{S}^{\mathsf{T}}\underline{d}'$ where

$$\underline{d}' \;=\; \begin{bmatrix} \underline{d}_L^{\mathsf{T}} & \underline{d}_R^{\mathsf{T}} & d_j \end{bmatrix}^{\mathsf{T}} = \begin{bmatrix} \underline{d}_U^{\mathsf{T}} & d_j \end{bmatrix}^{\mathsf{T}} .$$

The matrix $\underline{S} \in \{0, 1\}^{MK,MK}$ is defined as

$$\underline{S} = \begin{bmatrix} \underline{I}_j & 0 & \underline{0} \\ \underline{0} & 0 & \underline{I}_{MK-j-1} \\ 0 & 1 & 0 \end{bmatrix} ,$$

where post–multiplying any matrix $\underline{X}$ by $\underline{S}$ moves the right most column to column j, and pre–multiplication by $\underline{S}^{\mathsf{T}}$ moves the bottom row to row j. Also $\underline{S}$ has the following properties

$$\underline{S}\underline{S}^{\mathsf{T}} = \underline{S}^{\mathsf{T}}\underline{S} = \underline{I}.$$

Now forming the product $\underline{A}\underline{d}$ we get

$$\underline{A}\underline{d} = \underline{A}'\underline{S}\underline{S}^{\mathsf{T}}\underline{d}' = \underline{A}'\underline{d}'.$$

We then have the permuted channel model

$$\underline{e} = \begin{bmatrix} \underline{A}_U & \underline{a}_j \end{bmatrix} \begin{bmatrix} \underline{d}_U \\ d_j \end{bmatrix} + \underline{n}. \tag{1}$$

We can always arrange our permutation of $\underline{e}$ to have any transmission of interest (j) isolated algebraically at the bottom of the model. Although not absolutely necessary in the derivation of the projection receiver, the use of the $\underline{S}$ matrix makes the details simpler and is required in the comparison to the decorrelator. The hybrid constrained/unconstrained MLSE of $\underline{d}$ is now

$$\hat{\underline{d}} = \arg \min_{\hat{d}_j \in \mathcal{D}} \min_{\hat{\underline{d}}_U \in \mathbb{R}^{MK-1}} \left\| \underline{e} - \underline{A}\underline{d} \right\|^2. \tag{2}$$

The $\hat{\underline{d}}_U$ that satisfies the inner minimisation of (2) is

$$\hat{\underline{d}}_U = \left(\underline{A}_U^{\mathsf{T}}\underline{A}_U\right)^{-1} \underline{A}_U^{\mathsf{T}} \left(\underline{e} - \underline{a}_j \hat{d}_j\right).$$

The constrained minimisation problem for d_j is now

$$\begin{aligned} \hat{d}_j &= \arg \min_{\hat{d}_j \in \mathcal{D}} \left\| \left(\underline{I} - \underline{A}_U \left(\underline{A}_U^{\mathsf{T}}\underline{A}_U\right)^{-1} \underline{A}_U^{\mathsf{T}}\right) \left(\underline{e} - \underline{a}_j \hat{d}_j\right) \right\|^2 \\ &= \arg \min_{\hat{d}_j \in \mathcal{D}} \left\| \underline{M}_j \left(\underline{e} - \underline{a}_j \hat{d}_j\right) \right\|^2, \end{aligned} \tag{3}$$

where

$$\underline{M}_j = \underline{I} - \underline{A}_U \left(\underline{A}_U^{\mathsf{T}}\underline{A}_U\right)^{-1} \underline{A}_U^{\mathsf{T}}$$

is the projection matrix onto the null space of the space spanned by the columns of $\underline{A}_U$, i.e., null (span $\{\underline{A}_U\}$). The projection receiver first cancels the hypothesis it makes about transmission j and then projects out all other transmissions, hence the terminology projection receiver. Since the noise in $\underline{e}$ is white we can interpret (3) as the Euclidean metric for the hypothesis $\hat{d}_j$ corresponding to user $\kappa(j)$'s $\eta(j)^{\text{th}}$ symbol

$$\lambda_{PR,\kappa(j),\eta(j)} = \left\| \underline{M}_j (\underline{e} - \underline{a}_j \hat{d}_j) \right\|^2. \tag{4}$$

The $\hat{d}_j \in \mathcal{D}$ that minimises this metric would be judged to be the most likely. The metric for both hypotheses $(-1, +1)$ are placed in the metric stream for the user and passed on to the single user error control decoder.

3.2 Decorrelating Receiver

The well known decorrelating filter or decorrelator [2, 8] eliminates all of the multiuser interference in $\underline{e}$ at the expense of noise enhancement. The output of the decorrelator is

$$\underline{y}_{DC} = \left(\underline{A}^{\mathrm{T}}\underline{A}\right)^{-1}\underline{A}^{\mathrm{T}}\underline{e} = \underline{d} + \left(\underline{A}^{\mathrm{T}}\underline{A}\right)^{-1}\underline{A}^{\mathrm{T}}\underline{n} \in \mathbb{R}^{MK}.$$

A Euclidean metric for the hypothesis $\hat{d}_j$ based on the decorrelator statistic is

$$\lambda_{DC,\kappa(j),\eta(j)} = \left\| \left[\left(\underline{A}^{\mathrm{T}}\underline{A}\right)^{-1}\underline{A}^{\mathrm{T}}\underline{e}\right]_j - \hat{d}_j \right\|^2, \tag{5}$$

where $[\cdot]_j$ denotes element j of the vector.

3.3 Projection Receiver vs Decorrelating Receiver

For the purpose of comparison between the projection receiver and the decorrelator fed system let us consider component j of $\left(\underline{A}^{\mathrm{T}}\underline{A}\right)^{-1}\underline{A}^{\mathrm{T}}\underline{e}$ as required in (5).

$$\begin{aligned}
\left[\left(\underline{A}^{\mathrm{T}}\underline{A}\right)^{-1}\underline{A}^{\mathrm{T}}\underline{e}\right]_j &= \left[\left(\underline{S}^{\mathrm{T}}\underline{A}'^{\mathrm{T}}\underline{A}'\underline{S}\right)^{-1}\underline{S}^{\mathrm{T}}\underline{A}'^{\mathrm{T}}\underline{e}\right]_j \\
&= \left[\underline{S}^{\mathrm{T}}\left(\underline{A}'^{\mathrm{T}}\underline{A}'\right)^{-1}\underline{A}'^{\mathrm{T}}\underline{e}\right]_j \\
&= \left[\left(\underline{A}'^{\mathrm{T}}\underline{A}'\right)^{-1}\underline{A}'^{\mathrm{T}}\underline{e}\right]_{MK-1}
\end{aligned}$$

Using a well known matrix inversion lemma we have

$$\left(\underline{A}'^{\mathrm{T}}\underline{A}'\right)^{-1} = \begin{bmatrix} \left(\underline{A}_U^{\mathrm{T}}\underline{A}_U\right)^{-1}\left[\underline{I} + \rho_j\underline{A}_U^{\mathrm{T}}\underline{a}_j\underline{a}_j^{\mathrm{T}}\underline{A}_U\left(\underline{A}_U^{\mathrm{T}}\underline{A}_U\right)^{-1}\right] & -\rho_j\left(\underline{A}_U^{\mathrm{T}}\underline{A}_U\right)^{-1}\underline{A}_U^{\mathrm{T}}\underline{a}_j \\ -\rho_j\underline{a}_j^{\mathrm{T}}\underline{A}_U\left(\underline{A}_U^{\mathrm{T}}\underline{A}_U\right)^{-1} & \rho_j \end{bmatrix},$$

where $\rho_j = 1/\underline{a}_j^{\mathrm{T}}\underline{M}_j\underline{a}_j$. As well known for the decorrelator $\rho_j > 1/L$ and ρ_j increases as the channel becomes more strongly correlated. Also note the energy transmitted in one symbol interval by one user is $E_d = \underline{a}_j^{\mathrm{T}}\underline{a}_j$. We now have

$$\begin{aligned}
\left[\left(\underline{A}'^{\mathrm{T}}\underline{A}'\right)^{-1}\underline{A}'^{\mathrm{T}}\underline{e}\right]_{MK-1} &= \begin{bmatrix} -\rho_j\underline{a}_j^{\mathrm{T}}\underline{A}_U\left(\underline{A}_U^{\mathrm{T}}\underline{A}_U\right)^{-1} & \rho_j \end{bmatrix} \begin{bmatrix} \underline{A}_U^{\mathrm{T}} \\ \underline{a}_j^{\mathrm{T}} \end{bmatrix} \underline{e} \\
&= \left(-\rho_j\underline{a}_j^{\mathrm{T}}\underline{A}_U\left(\underline{A}_U^{\mathrm{T}}\underline{A}_U\right)^{-1}\underline{A}_U^{\mathrm{T}} + \rho_j\underline{a}_j^{\mathrm{T}}\right)\underline{e} \\
&= \rho_j\underline{a}_j^{\mathrm{T}}\underline{M}_j\underline{e}.
\end{aligned}$$

Finally we have the metric generated by the decorrelator for the hypothesis $\hat{d}_j$ in the form

$$\lambda_{DC,\kappa(j),\eta(j)} = \left\|\rho_j\underline{a}_j^{\mathrm{T}}\underline{M}_j\underline{e} - \hat{d}_j\right\|^2. \tag{6}$$

We are now in a position to compare the metric streams generated by the projection receiver and the decorrelator. This is of interest because it is these metrics that determine the performance of the system.

Lemma 1 The performance of the decorrelator and the projection receiver are identical in an uncoded communication system.

Proof: In uncoded operation hard decision are made on d_j as follows

$$\hat{d}_j = \arg\min_{\hat{d}_j \in \mathcal{D}} \lambda_{\kappa(j),\eta(j)} \left(\hat{d}_j\right).$$

When using metrics for hypothesis testing all terms in the metric that are independent of the particular hypothesis under consideration can be neglected. The decisions made by the projection receiver using (4) become

$$\begin{aligned} \hat{d}_j &= \arg\min_{\hat{d}_j \in \mathcal{D}} \lambda_{PR,\kappa(j),\eta(j)} \left(\hat{d}_j\right) \\ &= \arg\min_{\hat{d}_j \in \mathcal{D}} \left\| \underline{M}_j (\underline{e} - \underline{a}_j \hat{d}_j) \right\|^2 \\ &= \arg\min_{\hat{d}_j \in \mathcal{D}} -2\underline{e}^\mathsf{T} \underline{M}_j \underline{a}_j \hat{d}_j, \end{aligned}$$

and the decorrelator (6) decisions are

$$\begin{aligned} \hat{d}_j &= \arg\min_{\hat{d}_j \in \mathcal{D}} \lambda_{DC,\kappa(j),\eta(j)} \left(\hat{d}_j\right) \\ &= \arg\min_{\hat{d}_j \in \mathcal{D}} \left\| \rho_j \underline{a}_j^\mathsf{T} \underline{M}_j \underline{e} - \hat{d}_j \right\|^2 \\ &= \arg\min_{\hat{d}_j \in \mathcal{D}} -2\rho_j \underline{e}^\mathsf{T} \underline{M}_j \underline{a}_j \hat{d}_j. \end{aligned}$$

Now since ρ_j is a positive constant (w.r.t. the detection of d_j) the decisions made by the PR will be the same as the decisions made by the DC. ■

4 Decoding Performance Analysis

The branch metrics for the decoders may be developed as a sum of squared codeword bit distances. To be able to use the developed performance analysis for convolutional codes [9] we must derive an expression P_w for the probability that a Hamming weight w path ($\underline{y}^w$) has a better metric than the all zero path (or all -1 path) ($\underline{y}^0$), which we assume to be transmitted. To this end we define the metric for the path $\underline{y}$ as

$$\Gamma_k(\underline{y}) = \sum_{m=1}^{M_u} \sum_{i=1}^{n} \lambda_{k,(m-1)n+i} \left(y_{m,i}\right).$$

Let us study the performance of the decoders fed by the DC and PR metric streams.

4.1 Decorrelator fed BER

For the decoder of user k fed by the decorrelator metric stream,

$$\begin{aligned} P_{DC,w} &= P\left(\Gamma(\underline{y}^w) < \Gamma(\underline{y}^0)\right) \\ &= P\left(\sum_{i=1}^{w} \lambda_{DC,k,i}(1) < \sum_{i=1}^{w} \lambda_{DC,k,i}(-1)\right) \\ &= P\left(\sum_{i=1}^{w} \left(-1 + \rho_{\jmath(k,i)} \underline{a}_{\jmath(k,i)}^\mathsf{T} \underline{M}_{\jmath(k,i)} \underline{n}_{\jmath(k,i)}\right) > 0\right) \qquad (7) \\ &= P\left(\Omega_{DC,w} > 0\right), \end{aligned}$$

where

$$\Omega_{DC,w} \sim N\left(-w, \frac{N_0}{2}\sum_{i=1}^{w} \rho_{\jmath(k,i)}\right).$$

This then leads to the following probability,

$$P_{DC,w} = Q\left(\sqrt{\frac{2w^2}{N_0 \sum_{i=1}^{w} \rho_{\jmath(k,i)}}}\right). \tag{8}$$

4.2 Projection Receiver fed BER

The decoded path $\underline{\hat{y}}$ constructed by a decoder fed by the projection receiver metric stream is

$$\begin{aligned} \underline{\hat{y}} = \arg\min_{\underline{y}} \Gamma_{PR,k}\left(\underline{y}\right) &= \arg\min_{\underline{y}} \sum_{j} \lambda_{PR,\kappa(j),\eta(j)}\left(y_j\right) \\ &= \arg\min_{\underline{y}} \sum_{j} \rho_j^{-1} \lambda_{DC,\kappa(j),\eta(j)}\left(y_j\right). \end{aligned}$$

We can then get the probability P_w for the projection receiver from (7)

$$\begin{aligned} P_{PR,w} &= P\left(\sum_{i=1}^{w} \rho_{\jmath(k,i)}^{-1}\left(-1 + \rho_{\jmath(k,i)} \underline{a}_{\jmath(k,i)}^{\top} \underline{M}_{\jmath(k,i)} \underline{n}_{\jmath(k,i)}\right) > 0\right) \\ &= P\left(\Omega_{PR,w} > 0\right), \end{aligned}$$

where

$$\Omega_{PR,w} \sim N\left(-\sum_{i=1}^{w} \rho_{\jmath(k,i)}^{-1}, \frac{N_0}{2}\sum_{i=1}^{w} \rho_{\jmath(k,i)}^{-1}\right).$$

We therefore get

$$P_{PR,w} = Q\left(\sqrt{\frac{2\left(\sum_{i=1}^{w} \rho_{\jmath(k,i)}^{-1}\right)}{N_0}}\right). \tag{9}$$

Lemma 2 For an asynchronous, time varying CDMA channel $P_{PR,w} \leq P_{DC,w}$.

Proof: The arguments of the Q functions in (8) and (9) satisfy

$$\frac{1}{w}\sum_{i=1}^{w} \rho_{\jmath(k,i)}^{-1} \geq \frac{w}{\sum_{i=1}^{w} \rho_{\jmath(k,i)}}, \tag{10}$$

due to Jensen's inequality. Note that equality is only achieved when $\rho_{\jmath(k,m)} = \rho_k$. ■

Corollary 1 For a synchronous, time–invariant CDMA channel $P_{PR,w} = P_{DC,w}$.

Proof: When the users each pick a code and use that code for spreading for all time the synchronous channel has the property that $\rho_{\jmath(j,m)} = \rho_k$, thus equality is achieved in (10). ■

5 Simulation

In this section we simulate the PR and DC metric stream versions of the coded system on the random–code asynchronous and synchronous CDMA channel. The results are shown in Fig. 2 where the dotted curves pertain to a synchronous channel and the solid curves an asynchronous channel. The error control code used was a rate 1/2 code with constraint length $\nu = 3$, generators $G = 5, 7$ and $d_{\text{free}} = 5$. A system consisting of 5 users with spreading codes of length 7 is considered.

As was shown in Sec. 4 the performance of the PR should be better that the DC fed system. This is verified in Fig. 2. As is normally the case the code matched filter metric is not effective whereas the DC and PR metrics are combating the MUI, the PR more effectively than the DC. The performance for the single user system and the synchronous channel where each user transmits using a Gold code of length 7 are also shown. We see that a substantial price is paid for allowing asynchronous reception of the users where the network requirements are small. The advantage of the synchronous channel is that spreading code design is possible and the Gold codes are a good example. It is true however that the management of the synchronous channel with such special codes may negate the large benefits it has over the asynchronous channel.

6 Conclusion

We have derived a new multiuser linear complexity receiver and used it to supply metrics to soft input single user decoders in a coded asynchronous multiuser CDMA communications system. It was shown analytically that the projection receiver fed decoders will perform better than the decorrelator fed decoders. We have also seen that the PR and DC fed system will perform identically in the time–invariant synchronous channel.

References

[1] A. Duel-Hallen, "A family of multiuser decision–feedback detectors for asynchronous code–division multiple–access channels," *IEEE Transactions on Communications*, vol. 43, pp. 421–434, Feb. 1995.

[2] R. Lupas and S. Verdú, "Linear multiuser detectors for synchronous code–division multiple–access channels," *IEEE Transactions on Information Theory*, vol. 35, pp. 123–136, Jan. 1989.

[3] Z. Xie, C. K. Rushforth, and R. T. Short, "Multiuser signal detection using sequential decoding," *IEEE Transactions on Communications*, vol. 38, pp. 578–583, May 1990.

[4] M. K. Varanasi and B. Aazhang, "Multistage detection in asynchronous code–division multiple–access communications," *IEEE Transactions on Communications*, vol. 38, pp. 509–519, Apr. 1990.

[5] L. Wei and C. Schlegel, "Synchronous DS-SSMA with improved decorrelating decision–feedback multiuser detection," *IEEE Transactions on Vehicular Technology*, vol. 43, pp. 767–772, Aug. 1994. Special Issue on Future PCS Technologies.

[6] A. J. Grant, P. D. Alexander, and P. Whiting, “Why design spreading codes for multiuser CDMA channels?,” in *The Sixth International Symposium on Personal, Indoor, and Mobile Radio Communications*, Sept. 1995.

[7] C. Schlegel and Z.-J. Xiang, “Multiuser projection receivers,” in *IEEE International Symposium on Information Theory*, Sept. 1995.

[8] P. Jung and J. Blanz, “Joint detection with coherent receiver antenna diversity in CDMA mobile radio systems,” *IEEE Transactions on Vehicular Technology*, vol. 44, pp. 76–88, Feb. 1995.

[9] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*. Prentice Hall, 1995.

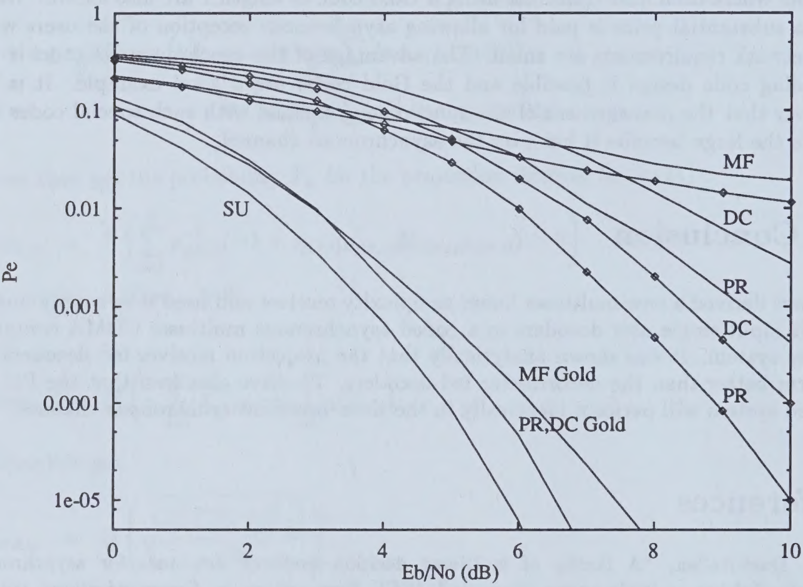


Figure 2: Decoder Performance for the Time–Varying Asynchronous CDMA Channel

Decoding Delay of Rate-Splitting Multiple Accessing with Bursty Sources *

S. P. Fedortsov and B. Rimoldi*
Washington University
Department of Electrical Engineering
Electronics Systems and Signals Research Laboratory
St. Louis, MO 63130, USA

I Abstract

A multi-user system with rate-splitting multiple-accessing is considered. In such a system there is a way to index users so that the code associated to transmitter i is a single-user code designed viewing the signal of transmitter j, $j > i$, as additive white Gaussian noise. The decoder does successive cancellation (also called onion peeling or stripping). This implies that the decoding delay of a given user depends on the number of users that need to be "canceled" before this user can be decoded. The decoding delay is analyzed for a certain class of bursty sources.

II Introduction

Rate-splitting multiple-accessing (RSMA) is a recently proposed multiple-access technique which allows one to achieve any point in the capacity region of a Gaussian multiple-access channel at comparably low complexity and without the need for synchronization among users [1].

Key to RSMA is the possibility for a user to use two (as opposed to one) codebooks. Conceptually, we think of each user as having two virtual transmitters. One can then show that, regardless of the power constraints and of the desired rate

---

*This work was supported by NSF Grant No. NCR-93-57689.

tuple (which is supposed to be in the capacity region), there is a way to assign powers and rates to virtual transmitters and to index them in such a way that the code associated to virtual transmitter i is a single-user code designed viewing the signal of virtual transmitter j, $j > i$, as additive white Gaussian noise. The decoder first decodes user 1 viewing the other signals as noise, subtract its contribution from the received signal, and proceeds with the next user until all users have been decoded. This procedure is called successive cancellation. In [1], as it is often the case, it was assumed that sources are infinite reservoirs of data made available at a constant rate.

This paper is a first attempt to study RSMA under the assumption of bursty sources. RSMA generalizes to arbitrary discrete memoryless multiple-access channels [2] and so does the result presented in this paper. A related work combining queueing theory with information theory for multiple accessing may be found in [3]

The Model

We assume an asynchronous discrete-time Gaussian multiple-access channel.[1] Loosely speaking, rate-splitting multiple-accessing [1] tells us that, without loss of generality, we may assume that user i sees users with higher index as noise. In order to be able to encode variable-length data bursts, we assume that transmitter i has a family of codes, all codewords of the family having the same rate R_i but various codeword-lengths. In order to describe the decoding operation, it is convenient to envision the sequence of codewords transmitted by user i as laying on top of the sequence transmitted by user $i+1$, as shown in Fig. 1. According to our index convention, codewords at lower level are considered as noise by codewords at higher levels. The actual noise (considered as noise by all users) is shown at the bottom of Fig. 1. We assume that the rate R_i of layer i is kept constant. This is necessary if there is no feedback since then transmitter i will never know the traffic of lower layers and cannot adapt his rate to it.

Given the random process describing source i, and given a rule on how to partition the source output into packets of data that are encoded separately, one can (in principle) determine a stochastic process that describes the temporal location of codewords (Fig. 1). In this paper we make the simplistic assumption that codeword lengths of level i are independent and identically distributed (i.i.d.) random variables with mean η_i and second moment $\eta_i^{(2)}$, and that inter-codeword distances are i.i.d. with mean τ_i and independent of the codeword lengths. In order to avoid transients, we assume that the state of the system at initial time moment $t = 0$ is a probabilistic replica of its asymptotic state as $t \to \infty$. With this assumption, the expected decoding delay of any packet at level i is independent of time.

---

[1] We choose a discrete-time channel model for notational simplicity only. Our result generalizes to continuous-time Gaussian multiple-access channels.

Pessimistically, we assume that a codeword can be decoded if and only if it does not overlap in time with a codeword at higher levels. If this condition is satisfied, the codeword under consideration is said to be *free.* A free codeword is decoded at once and, conceptually, removed from Fig. 1. We assume that the actual decoding operation of a free codeword takes no time. In Fig. 1 the decoding order is a, b, c, d, e, f, g.

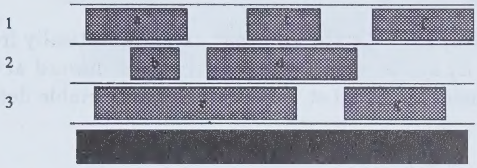


Figure 1: Layered channel representation. Codewords are represented in gray and the additive white Gaussian noise is represented in black.

Our main interest is in characterizing the contributions to the transmission delay which are peculiar to RSMA systems. Of primary interest is the *waiting time*, defined as the time elapsed from the end of a codeword until that codeword becomes free and, thus, it is decoded.

It should be pointed out that in a more general situation the instantaneous rate R_i of user i might be negotiable between this user and, say, the common receiver. The tradeoff is that higher layers have lower rate but also smaller waiting time and, presumably, lower utilization fee. Since lower rates means longer codewords and, in turn, larger delays, the length of the codeword should be added to the delay. For the model considered in this paper the average codeword length of each layer is given and, thus, its computation is not an issue (see also the examples at the end).

B. Decoding Delay

In this section we determine the average waiting time ω_i of a codeword at level i.

Theorem 1 Assume that the sequence of codeword lengths of level i is i.i.d. with mean η_i and second moment $\eta_i^{(2)}$, that the sequence of inter-codeword distances is i.i.d. with mean τ_i, and that levels are independent. Then the mean waiting time ω_i is given by the recursion

$$\omega_i = \begin{cases} 0, & i = 1 \\ \sum_{j=1}^{i-1} \left[\frac{\eta_j^{(2)}}{2\eta_j} + \omega_j \right] \pi_j \prod_{l=j+1}^{i-1} (1 - \pi_l), & i = 2, \cdots, m, \end{cases} \tag{1}$$

where

$$\pi_l = \frac{\eta_l}{\eta_l + \tau_l}.$$

Proof. Fix a codeword at level i, such as codeword g in Fig. 2. This is THE codeword for which we will compute the expected waiting time ω_i.

Toward this goal, let V be the line that extends vertically from the end of the codeword. Define E_j as the event that V crosses a codeword at level j, $j < i$ and denote its complement by E_j^c . Let J be the random variable defined by

$$J = \begin{cases} 0, & \text{if the event } \cap_{l=1}^{i-1} E_l^c \text{ occurs} \\ j, & \text{if the event} E_j \cap_{l=j+1}^{i-1} E_l^c \text{ occurs.} \end{cases}$$

In words, J is the largest index j, $j < i$, such that a codeword at level j is crossed by V and is 0 if no such crossing occurs.

The key observation is that the codeword becomes free as soon as the codeword crossed by V at level J becomes free (codeword c in Fig. 2). To see why, first notice that until c is removed g is not free. On the other hand in order for c to be free there can not be any (piece of) codeword on top of c. In particular, after c is removed, the vertical line that extends vertically above the end of codeword g can not cross any codeword. This means that all codewords to the left of this line and at levels above level i are free.

If V does not intersect any codeword then the codeword is free upon reception. Define

$$W_i = \begin{cases} 0, & J = 0 \\ W_J + Z, & \text{otherwise,} \end{cases}$$

where Z is defined as in Fig. 2.

Let N_J be the length of the codeword intercepted by V at level J and define

$$\omega_i(j, n) := E\{W_i | J = j, N_j = n\}. \tag{2}$$

Under the condition $J = j$ and $N_j = n$, the random variable Z is uniformly distributed in $[0, n]$. Hence

$$\omega_i(j, n) = E[W_j + Z | J = j, N_j = n] = \omega_j + \frac{n}{2}$$

where we made use of the fact that layers are independent and in order to avoid making a special case for $j = 0$ we assume that $N_0 = 0$ with probability one.

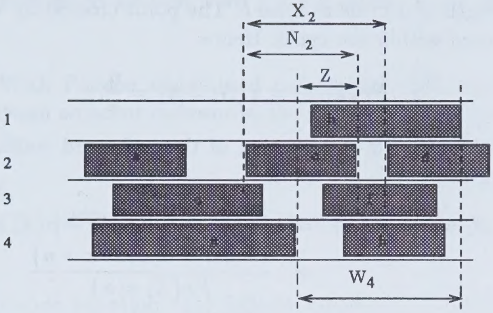


Figure 2: Aggregate variables.

Now we proceed to determine the probability distribution $Pr\{J = j, N_j = n\}$.

$$\begin{aligned} Pr\{J = j, N_j = n\} &= Pr\{N_j = n, E_{i-1}^c, E_{i-2}^c, \cdots, E_j\} \\ &= Pr\{N_j = n | E_{i-1}^c, E_{i-2}^c, \cdots, E_j\} Pr\{E_{i-1}^c, E_{i-2}^c, \cdots, E_j\} \\ &= Pr\{N_j = n | E_j\} \prod_{l=j+1}^{i-1} (1 - Pr\{E_l\}) Pr\{E_j\}. \end{aligned} \tag{3}$$

Now we proceed to prove the equality

$$Pr\{E_l\} = \pi_l.$$

Toward this goal define a *cycle* of level l as an interval consisting of a codeword of level l and the subsequent waiting time. THE cycle of level l the one which is crossed by V. Let T_l be the inter-codeword distance and $X_l = N_l + T_l$ be the length of the cycle of level l. Denote by $p_l(k)$ the probability mass function of N_l and by $q_l(x)$ the probability mass function of T_l. Notice that while an arbitrary cycle at level l has length distributed according

$$g_l(x) = \sum_{n=0}^{x} p_l(n) q_l(x - n),$$

longer cycles are more likely to be crossed by V than shorter ones. The appropriate probability mass function for THE cycle of level l is [4]

$$f(x) = \frac{x g_l(x)}{\xi_l}$$

where

$$\xi_l = \eta_l + \tau_l$$

is the expected length of a cycle at level l. The point crossed by V in THE cycle is uniformly distributed within the cycle. Hence

$$Pr\{E_l | X_l = x, N_l = n\} = \frac{n}{x}. \tag{4}$$

For a given cycle length, the distribution of the codeword contained in it can be determined as follows

$$\begin{aligned} Pr\{N_l = n | X_l = x\} &= Pr\{N_l = n, T_l = x - n | X_l = x\} \\ &= \frac{Pr\{N_l = n, T_l = x - n\}}{Pr\{X_l = x\}} \\ &= \frac{p_l(n) q_l(x-n)}{g_l(x)}. \end{aligned}$$

Hence

$$Pr\{N_l = n, X_l = x\} = p_l(n) q_l(x-n) \frac{x}{\xi_l}. \tag{5}$$

From (5) and (4) we obtain

$$\pi_l = \frac{1}{\xi_l} \sum_{n=0}^{\infty} n p_l(n) \sum_{r=0}^{\infty} q_l(r) = \frac{\eta_l}{\xi_l} = \frac{\eta_l}{\eta_l + \tau_l}. \tag{6}$$

Hence the probability π_l that V crosses a codeword at level l is the average codeword length divided by the average cycle length: an intuitive result.

Next we determine $Pr\{N_j = n | E_j\}$.

$$\begin{aligned} Pr\{N_j = n | E_j\} &= \frac{1}{\pi_j} \sum_{x=n}^{\infty} Pr\{N_j = n, X_j = x, E_j\} \\ &= \frac{1}{\pi_j} \sum_{x=n}^{\infty} Pr\{E_j | N_j = n, X_j = x\} Pr\{N_j = n, X_j = x\} \\ &= \frac{n p_j(n)}{\eta_j}, \end{aligned} \tag{7}$$

where we made use of (4), (5), and (6).

Finally, using (7), (3), and (2) we obtain

$$\begin{aligned} \omega_i &= \sum_{j=0}^{i-1} \sum_{n=0}^{\infty} \omega_i(j,n) Pr\{J = j, N_j = n\} & (8) \\ &= \sum_{j=0}^{i-1} \pi_j \prod_{l=j+1}^{i-1} (1 - \pi_l) \sum_{n=0}^{\infty} \left(\frac{n}{2} + \omega_j \right) \frac{n p_j(n)}{\eta_j} & (9) \\ &= \sum_{j=1}^{i-1} \left[\frac{\eta_j^{(2)}}{2\eta_j} + \omega_j \right] \pi_j \prod_{l=j+1}^{i-1} (1 - \pi_l), & (10) \end{aligned}$$

where in the last expression we sum from $j = 1$ since for $j = 0$ the summand is 0. □

Corollary. With Poisson distributed codeword lengths and arbitrarily distributed times between adjacent codewords, the mean waiting time ω_i of a codeword at level i is given by

$$\omega_i = \begin{cases} 0, & i = 1 \\ \sum_{j=1}^{i-1} \left[\frac{(1+\eta_j)}{2} + \omega_j\right] \pi_j \prod_{l=j+1}^{i-1}(1-\pi_l), & i = 2, \cdots, m, \end{cases} \tag{11}$$

Proof. Recurrence equation (11) follows directly from (1) since the second moment of a Poisson distributed RV with parameter η equals $\eta(1+\eta)$. □

III Example

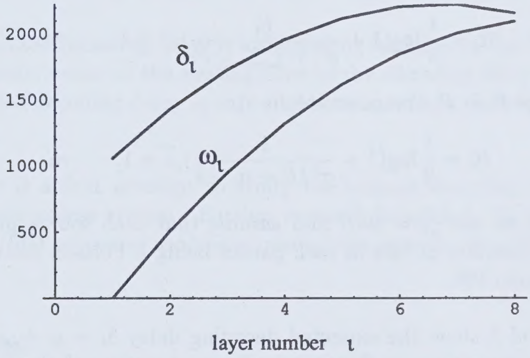


Figure 3: The inter-codeword distance is equal to 50.

In this example we assume that the noise has power σ^2 and that user i has power P_i. Furthermore, we assume that each transmitter uses ideal codes that achieve

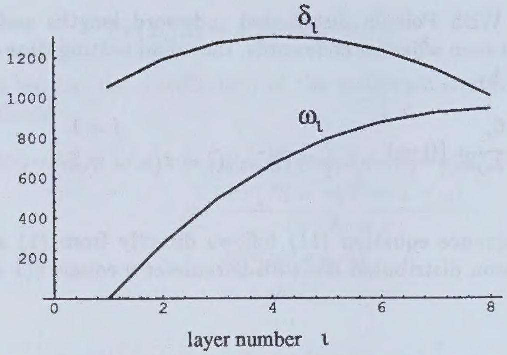


Figure 4: Inter-codeword distance is equal to 1000.

capacity on the single-user channel seen by each individual user. Hence

$$R_i = \frac{1}{2}\log(1 + \frac{P_i}{\sigma^2 + \sum_{j=i+1}^{m} P_i}), i = 1, \cdots, m. \tag{12}$$

For equal powers $P_i = P$, the rates are

$$R_i = \frac{1}{2}\log(1 + \frac{1}{\sigma^2/P + m - i}), i = 1, \cdots, m. \tag{13}$$

As an example, we use $\frac{P}{\sigma^2} = 9dB$ and assume that each source produces packets of bits with the number of bits in each packet being a Poisson distributed random variable with mean 100.

Fig. 3, 4 and 5 show the expected decoding delay $\delta_i = \eta_i + \omega_i$ (upper curve) and expected waiting time ω_i (lower curve) as a function of the level number for $\tau_i = \tau$ equal 50, 1000, and 5000, respectively. For illustration purposes, we have interpolated between points.

As one expects, the expected waiting time is an increasing function of the level number. Moreover, if all levels have the same transmission rate $R_i = R$ (which is not the case in our example), the average decoding delay as well as the average waiting time increase linearly with the level number. It is perhaps surprising that in general, the average decoding delay of a lower level can be less than that of an upper level (see Fig. 4). If the expected inter-codeword distance is large, as for example in

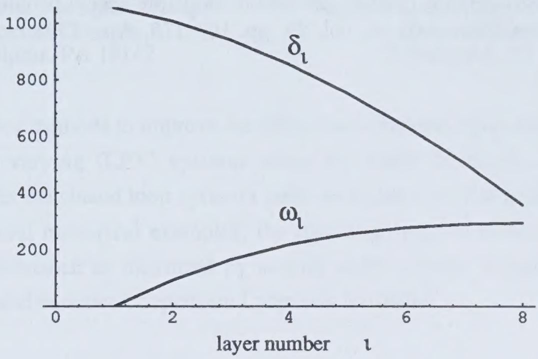


Figure 5: Inter-codeword distance is equal to 5000.

Fig. 5, the expected decoding delay is a decreasing function of the level number. In this case the contribution of the waiting time to the decoding delay is very small so that the expected decoding delay is only slightly larger than the expected codeword length.

This paper is a first attempt to study the average decoding delay of a rate-splitting multiple-access system. Further research is needed. In particular, more elaborate busy/idle processes modeling transmitter outputs should be considered.

References

[1] B. Rimoldi and R. Urbanke, "A rate-splitting approach to the Gaussian multiple-access channel," *IEEE Trans. Inform. Theory.* Accepted for publication.

[2] A. Grant, B. Rimoldi, R. Urbanke, and P. Whiting, "On single-user coding for the discrete memoryless multiple-access channel," in *Proceedings 1995 IEEE International Symposium on Information Theor*, (Whistler, B.C. Canada), p. 448, September 17-22, 1995.

[3] İ. E. Telatar and R. G. Gallager, "Combining queueing theory with information theory for multiaccess," *IEEE J. Select. Areas Commun.*, vol. SAC–13, pp. 963–969, Aug. 1995.

[4] B. S. Tsybakov, "On a random process in multiple access problems," *Problems of Information Transmission*, vol. 29, pp. 163–175, Apr. 1993.

Improving the Robustness of Linear Parameter Varying Systems Using Parameter Feedback

David Malloy
Boeing Defense and Space Group
Helicopters Division
Philadelphia, PA 19142

B.C. Chang
Department of Mechanical Engineering
Drexel University
Philadelphia, Pa 19104

Abstract

This paper describes methods to improve the robustness of single-input, single-output (SISO) linear, parameter-varying (LPV) systems using parameter feedback. The feedback is structured to render the closed loop system's poles to be parameter independent. Using this technique on several numerical examples, the resulting systems exhibit improved robust stability and performance as measured by several stability tests including the structured singular value, μ, and parameter dependent Lyapunov functions.

Introduction

Parameter dependent linear systems are suitable mathematical models for various applications. As discussed in [10], linear parameter varying systems result from linearizing nonlinear systems about various operating points. One example, described in [13], are aircraft linear equations of motion that vary with dynamic pressure. Gain scheduling has been the traditional approach to control system design for parameter dependent systems, although there have been challenges associated with analyzing the stability of the resulting closed loop system [6], [11], [12]. Recent developments in assessing stability of LPV systems include those described in [8] and [3] which utilize quadratic Lyapunov functions. Also, design methods recently introduced utilize both linear fractional transformations and quadratic Lyapunov functions [9], [2], [7], [3].

A general LPV control problem can be described as follows. Consider a generalized linear system of the form

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}(\theta(t))\mathbf{x}(t)+\mathbf{B}_1(\theta(t))\mathbf{w}(t)+\mathbf{B}_2(\theta(t))\mathbf{u}(t)\\ \mathbf{z}(t) &= \mathbf{C}_1(\theta(t))\mathbf{x}(t)+\mathbf{D}_{11}(\theta(t))\mathbf{w}(t)+\mathbf{D}_{12}(\theta(t))\mathbf{u}(t)\\ \mathbf{y}(t) &= \mathbf{C}_2(\theta(t))\mathbf{x}(t)+\mathbf{D}_{21}(\theta(t))\mathbf{w}(t)+\mathbf{D}_{22}(\theta(t))\mathbf{u}(t)\end{aligned} \quad \text{or} \quad \left[\begin{array}{c|cc}\mathbf{A}(\theta(t)) & \mathbf{B}_1(\theta(t)) & \mathbf{B}_2(\theta(t))\\ \hline \mathbf{C}_1(\theta(t)) & \mathbf{D}_{11}(\theta(t)) & \mathbf{D}_{12}(\theta(t))\\ \mathbf{C}_2(\theta(t)) & \mathbf{D}_{21}(\theta(t)) & \mathbf{D}_{22}(\theta(t))\end{array}\right] \quad (1)$$

where $\theta(t)=\left[\theta_1(t),\theta_2(t),\dots,\theta_m(t)\right]$ are time varying parameters with bounded magnitude, $\mathbf{x}\in R^n$ are the system states, $\mathbf{z}$ are the weighted outputs that generally include error signals and control signals, $\mathbf{y}$ are the measured outputs, $\mathbf{u}$ are the control inputs and $\mathbf{w}$ are the exogenous inputs which include commands, perturbations and disturbances. The goal is to find a control law for $\mathbf{u}$ that minimizes the response of the weighted outputs, $\mathbf{z}$ to the exogenous inputs, $\mathbf{w}$.

The technique described in this paper uses parameter feedback to completely eliminate or minimize the size of the parameter dependent terms from the closed loop system's eigenvalues. The parameter feedback does not guarantee stability for all parameter variations but, in numerous numerical examples, we have observed marked improvements in system robustness as measured by the structured singular value, μ, as well as with parameter dependent Lyapunov function based tests. Straightforward application of pole placement algorithms for SISO systems implemented using a symbolic programming language [4] enable state feedback controllers, observer based controllers and output feedback controllers to achieve closed loop parameter independent poles. In this paper, emphasis will be placed on the state feedback case, due to its relative simplicity. The design concepts used for state feedback are directly applicable to improving the robustness of observer based controllers and output feedback controllers. This paper first outlines the stability tests used and then describes the parameter feedback design methods with examples.

Stability of Linear Parameter-Varying Systems

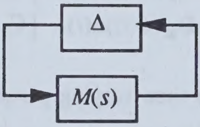
Structured Singular Value Test: A parameter dependent system can be approximated by a perturbed linear time invariant (LTI) system. For example, the LPV system

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\theta(t))\mathbf{x}(t) \qquad (2)$$

may be modeled as the LTI system

$$\dot{\mathbf{x}}(t) = \tilde{\mathbf{A}}\mathbf{x}(t) + \hat{\mathbf{A}}\Delta\mathbf{x}(t) \qquad (3)$$

in the region where the parameter varying matrix may be approximated by $\mathbf{A}(\theta(t)) \cong \tilde{\mathbf{A}} + \hat{\mathbf{A}}\Delta$. The matrices $\tilde{\mathbf{A}}$ and $\hat{\mathbf{A}}$ are constant, with $\hat{\mathbf{A}}$ scaled such that the perturbation block, Δ, has a norm satisfying $\|\Delta\| \leq 1$ for the range of interest of $\theta(t)$. If the parameter may be modeled as a time-invariant uncertainty, the stability of the LPV system (2) may be determined using the structured singular value, μ, in the region where the approximation (3) is valid . Let $M(s)$ be defined as the unperturbed LTI system $M(s) = \left[\begin{array}{c|c} \tilde{\mathbf{A}} & \hat{\mathbf{A}} \\ \hline \mathbf{I} & \mathbf{0} \end{array}\right]$. Then (3) is equivalent to the $M\Delta$ structure



The structured singular value μ is defined [1] as $\mu(M) = \{min\bar{\sigma}(\Delta) : \det(I - M\Delta) = 0\}^{-1}$. With $\|\Delta\| \leq 1$, the norm of the complex structured singular value of the $M\Delta$ system must satisfy $\|\mu(M)\| < 1$ for robust stability. An upper bound for μ, μ_{ub}, can be computed by solving the following linear matrix inequality (LMI) described in [5] and implemented in [7] and [1]:

$$\mu_{ub}(\mathbf{M}) = \sqrt{\max(0,\alpha^*)}$$
$$\mathbf{M}^{H}\mathbf{D}\mathbf{M} + j(\mathbf{G}\mathbf{M} - \mathbf{M}^{H}\mathbf{G}) < \alpha\mathbf{D}$$
$$\mathbf{D} > \mathbf{I}$$

where α^* is the global minimum found by minimizing α over the sets of scaling matrices $\mathbf{D} \in \mathcal{D}$ and $\mathbf{G} \in \mathcal{G}$. The sets of allowable scaling matrices are determined by the perturbation block's structure and whether the block is complex or real valued. If $\Delta = diag(\Delta_1, \Delta_2, \ldots, \Delta_n)$, then $\mathbf{D} = diag(\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_n)$ and $\mathbf{G} = diag(\mathbf{G}_1, \mathbf{G}_2, \ldots, \mathbf{G}_n)$. Also, $\mathbf{D}_i = d_i \,\mathrm{x}\, \mathbf{I}$ and $d_i > 0$ if Δ_i is a full block, and $\mathbf{D}_i = \mathbf{D}_i^H$ otherwise. The submatrices of $\mathbf{G}$ are defined as $\mathbf{G}_i = 0$ if Δ_i is complex-valued, $\mathbf{G}_i = g_i \,\mathrm{x}\, \mathbf{I}$ with $g_i \in R$ if Δ_i is a real-valued full block, and $\mathbf{G}_i = \mathbf{G}_i^H$ if $\Delta_i = \delta_i \,\mathrm{x}\, \mathbf{I}$ with δ_i real.

Parameter Dependent Lyapunov Function Stability Tests : As noted in [7], the LPV system (2) may be described or approximated as an affine parameter dependent system with a dynamics matrix such as $\mathbf{A}(\theta(t)) \cong \mathbf{A}_0 + \mathbf{A}_1\theta_1 + \mathbf{A}_2\theta_2 + \cdots + \mathbf{A}_m\theta_m$ where θ_i are independent parameters contained in the vector $\theta = [\theta_1, \theta_2, \ldots, \theta_m]$. The stability of the LPV system may be established by finding a parameter dependent Lyapunov function $V(x, \theta(t))$ such that (dropping the time dependence from the notation) [8]

$$V(x,\theta) = x^T\mathbf{P}(\theta)x > 0$$
$$\text{and} \quad \frac{dV(x,\theta)}{dt} = x^T\left(\mathbf{A}(\theta)^T\mathbf{P}(\theta) + \mathbf{P}(\theta)\mathbf{A}(\theta) + \frac{d\mathbf{P}(\theta)}{dt}\right)x \le 0$$
$$\text{where} \quad \mathbf{P}(\theta) = \mathbf{P}_0 + \mathbf{P}_1\theta_1 + \mathbf{P}_2\theta_2 + \cdots + \mathbf{P}_m\theta_m .$$

If the parameters have extreme values $\theta_i = [\underline{\theta_i}, \overline{\theta_i}]$, the parameter box may be defined as the set of vertices $V = \{\omega = (\omega_1, \ldots, \omega_m) : \omega_i \in \{\underline{\theta_i}, \overline{\theta_i}\}\}$. Similarly, if the rates of the parameters have extreme values $\dot{\theta}_i = [\underline{\dot{\theta}_i}, \overline{\dot{\theta}_i}]$, the parameter rate box may be defined by the set of vertices $T = \{\tau = (\tau_1, \ldots, \tau_m) : \tau_i \in \{\underline{\dot{\theta}_i}, \overline{\dot{\theta}_i}\}\}$. Then, if the following linear matrix inequality is feasible, it may be solved for $\mathbf{P}(\theta)$ from and the LPV system is stable for time varying θ:

$$\mathbf{A}(\omega)^T\mathbf{P}(\omega) + \mathbf{P}(\omega)\mathbf{A}(\omega) + \mathbf{P}(\tau) < \mathbf{P}_0 \quad \forall(\omega,\tau) \in V \,\mathrm{x}\, T$$
$$\mathbf{P}(\omega) > \mathbf{I} \quad \forall\omega \in V \tag{4}$$
$$\mathbf{A}_i^T\mathbf{P}_i + \mathbf{P}_i\mathbf{A}_i \ge 0 \quad for\ i = 1, \ldots, m$$

The last condition ensures convexity. If the parameters are infinitely time varying, $\dot{\theta}_i = [-\infty, \infty]$, then for feasibility $\mathbf{P}_1 = \mathbf{P}_2 = \cdots = \mathbf{P}_m = 0$ and (4) reduces to an LMI for quadratic stability.

Design Approaches

State Feedback: Given the system

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\theta(t))\mathbf{x}(t) + \mathbf{B}(\theta(t))\mathbf{u}(t)$$

Assume that $\{\mathbf{A}(\theta(t)),\mathbf{B}(\theta(t))\}$ is controllable for all $\theta(t)$ within the desired operating range and the measured value of the parameters is $\overline{\theta}(t)$. Then, using the measured parameter values, a feedback

$$\mathbf{u}(t)=\mathbf{F}(\overline{\theta}(t))\mathbf{x}(t) \tag{5}$$

may be found to

i) Eliminate the parameters from the closed loop eigenvalues when $\overline{\theta}(t)=\theta(t)$

ii) Reduce the size of the parameter dependent terms in the closed loop characteristic equation when $\overline{\theta}(t)\neq\theta(t)$.

iii) Possibly improve the closed loop system robust stability.

Suppose the desired characteristic equation of the closed loop system for frozen parameter values is

$$a(s)=s^n+a_{n-1}s^{n-1}+\ldots+a_0. \tag{6}$$

With

$$\mathbf{F}(\overline{\theta}(t))=\begin{bmatrix} f_1(\overline{\theta}(t)) & f_2(\overline{\theta}(t)) & \cdots & f_n(\overline{\theta}(t))\end{bmatrix} \tag{7}$$

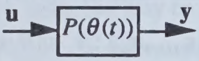
the resulting characteristic equation for $\mathbf{A}(\theta(t))+\mathbf{B}(\theta(t))\mathbf{F}(\overline{\theta}(t))$ for fixed parameter values, $\theta(t)=\theta$, is computed as

$$\tilde{a}(s,\theta,\overline{\theta},f_i)=s^n+\tilde{a}_{n-1}(\theta,\overline{\theta},f_i)s^{n-1}+\ldots+\tilde{a}_0(\theta,\overline{\theta},f_i). \tag{8}$$

Setting $\theta(t)=\overline{\theta}(t)$ and equating coefficients of equal powers of s from (6) and (8) leads to n linear equations with n unknowns, f_i. Solving this set of equations results in expressions for f_i that may be nonlinear with respect to $\overline{\theta}(t)$. The eigenvalues of the closed loop system $\mathbf{A}(\theta(t))+\mathbf{B}(\theta(t))\mathbf{F}(\overline{\theta}(t))$ will be parameter independent when $\overline{\theta}(t)=\theta(t)$. Our observation is that the closed loop robustness decreases as the difference between the measured and actual parameter value, $\left|\overline{\theta}(t)-\theta(t)\right|$, increases.

Example 1: The following simple example illustrates the state feedback method. The LPV system, $P(\theta(t))$, similar to the example in [12], is unstable when $\theta(t)=\cos 2t$.

$$P(\theta(t))=\left[\begin{array}{c|c}\mathbf{A}(\theta(t)) & \mathbf{B}\\ \hline \mathbf{C}(\theta(t)) & 0\end{array}\right]=\left[\begin{array}{cc|c}0 & 1 & 1\\ -1-.5\theta(t) & -.2 & -5\\ \hline -9+.6\theta(t) & 1 & 0\end{array}\right] \quad \text{or}$$



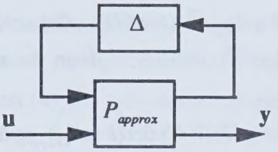
The frozen parameter characteristic polynomial is

$$s^2+.2s+1+.5\theta \quad \text{or} \quad s^2+.2s+1+\phi(\theta) \quad \text{with} \quad \phi(\theta)=.5\theta\,. \tag{9}$$

The design goal is to eliminate the parameter dependent term by using state feedback so that the closed loop characteristic polynomial is

$$s^2+.2s+1 \tag{10}$$

The open loop system, $P(\theta(t))$, is non-minimum phase, therefore precluding inverting the plant in order to cancel its dynamics including the effect of the parameters. For a frozen parameter value of $\theta=-50$, the system is uncontrollable. $P(\theta(t))$ may be approximated by the $P_{approx}\Delta$ structure



where

$$P_{approx} = \left[\begin{array}{c|c:c} \tilde{\mathbf{A}} & \hat{\mathbf{A}} & \mathbf{B} \\ \hline \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \hdashline \tilde{\mathbf{C}} & \hat{\mathbf{C}} & \mathbf{0} \end{array}\right] = \left[\begin{array}{cc|cc:c} 0 & 1 & 0 & 0 & 1 \\ -1 & -.2 & -.5 & 0 & -5 \\ \hline 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \hdashline -9 & 1 & .6 & 0 & 0 \end{array}\right] \quad \text{and} \quad \begin{array}{l} \mathbf{A}(\theta(t)) \cong \tilde{\mathbf{A}} + \hat{\mathbf{A}}\Delta \\ \mathbf{C}(\theta(t)) \cong \tilde{\mathbf{C}} + \hat{\mathbf{C}}\Delta \\ \Delta = \begin{bmatrix} \delta & 0 \\ 0 & \delta \end{bmatrix} \text{ with } |\delta| \le 1 \end{array}$$

This approximation facilitates stability analysis for the open loop LPV in the region of interest for $\theta(t)$, $|\theta(t)| \le 1$, by computing the $P_{approx}\Delta$ system's robust stability using μ such that

$$\mu(P_{approx}) = \{min\bar{\sigma}(\Delta) : \det(I - P_{approx}\Delta) = 0\}^{-1}$$

The state feedback gain,

$$F(\bar{\theta}(t)) = \begin{bmatrix} \dfrac{-5\bar{\theta}(t)}{50+\bar{\theta}(t)} & \dfrac{-\bar{\theta}(t)}{50+\bar{\theta}(t)} \end{bmatrix} \tag{11}$$

results in the closed loop system

$$P_{CL}(\theta(t),\bar{\theta}(t)) = \left[\begin{array}{cc|c} \dfrac{-5\bar{\theta}(t)}{50+\bar{\theta}(t)} & 1-\dfrac{\bar{\theta}(t)}{50+\bar{\theta}(t)} & 1 \\ -1-.5\theta(t)+\dfrac{25\bar{\theta}(t)}{50+\bar{\theta}(t)} & -.2+\dfrac{5\bar{\theta}(t)}{50+\bar{\theta}(t)} & -5 \\ \hline -9+.6\theta(t) & 1 & 0 \end{array}\right],$$

with the frozen parameter characteristic polynomial,

$$s^2+.2s+1+\frac{25}{50+\bar{\theta}}\left(\theta-\bar{\theta}\right) \quad \text{or} \quad s^2+.2s+1+\phi_{CL}(\theta,\bar{\theta}) \tag{12}$$

with
$$\phi_{CL}(\theta,\bar{\theta}) = \frac{25}{50+\bar{\theta}}\left(\theta-\bar{\theta}\right).$$

When the measured parameter value, $\bar{\theta}(t)$, equals the actual parameter value, $\theta(t)$, the parameter dependent term $\phi_{CL}(\theta,\bar{\theta}) = 0$, achieving the closed loop polynomial $s^2+.2s+1$.

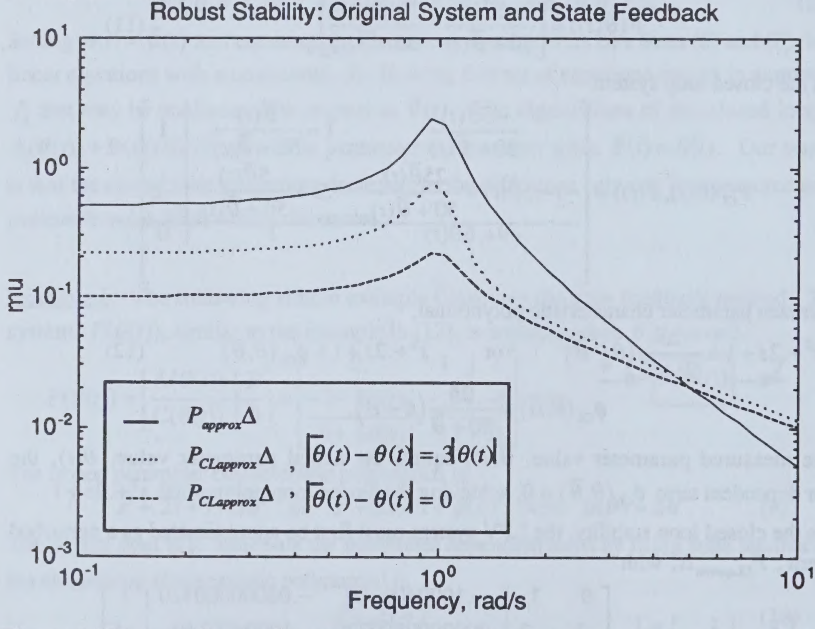
To assess the closed loop stability, the LPV system must first be approximated as a perturbed LTI system , $P_{CLapprox}\Delta$, with

$$P_{CLapprox} = \left[\begin{array}{c|c:c} \tilde{\mathbf{A}}_{CL} & \hat{\mathbf{A}}_{CL} & \mathbf{B} \\ \hline \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \hdashline \tilde{\mathbf{C}} & \hat{\mathbf{C}} & \mathbf{0} \end{array}\right] = \left[\begin{array}{cc|cc:c} 0 & 1 & -.1000400160 & -.02000800320 & 1 \\ -1 & -.2 & .0002000800320 & .1000400160 & -5 \\ \hline 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \hdashline -9 & 1 & .6 & 0 & 0 \end{array}\right]$$

and $\mathbf{A}_{CL}(\theta(t)) \cong \tilde{\mathbf{A}}_{CL} + \hat{\mathbf{A}}_{CL}\Delta$. The elements of the matrix $\hat{\mathbf{A}}_{CL}$ are the simply the average slopes of the linear approximations of the corresponding parameter dependent terms from $\mathbf{A}(\theta(t)) + \mathbf{B}(\theta(t))\mathbf{F}(\bar{\theta}(t))$ computed over the range of interest for $\theta(t)$. The particular values shown above for $\hat{\mathbf{A}}_{CL}$ were found using $\bar{\theta}(t)=\theta(t)$. Other values must be computed for $\hat{\mathbf{A}}_{CL}$ to assess robust stability when $\bar{\theta}(t)\neq\theta(t)$. Given the appropriate $P_{CLapprox}\Delta$ model, robust stability can be analyzed using

$$\mu(P_{CLapprox}) = \{min\bar{\sigma}(\Delta):\det(I - P_{CLapprox}\Delta) = 0\}^{-1}$$

Note that $\tilde{\mathbf{A}}_{CL}=\tilde{\mathbf{A}}$ because the feedback gain $F(\bar{\theta}(t))$ canceled only the parameter dependent terms in the closed loop characteristic equation and did not alter the parameter independent terms. Therefore, comparing the μ plots for robust stability for $P_{CLapprox}\Delta$ and $P_{approx}\Delta$ highlights the benefits of eliminating the parameter from the frozen closed loop system's eigenvalues, as shown below. The plots are for complex, time-invariant perturbations or variations in the parameter. Significant improvements in the system's robust stability are evident, even with large errors in parameter measurement. If $\bar{\theta}(t)$, is sufficiently close to or equal to, $\theta(t)$, $|\phi_{CL}(\theta,\bar{\theta})|<|\phi(\theta)|$ then the closed loop robust stability will be better than the open loop robust stability. Simulations confirm that the closed loop system has improved stability as compared to the original system.



The stability margins under strictly real perturbations or parameter variations are $1/\mu_{ub\,OL}=2.00$ and $1/\mu_{ub\,SF}=9.99$, for the open loop and parameter feedback cases, respectively. The parameter feedback improves the robustness against time-invariant, real parameter variations by a factor of nearly 5.

The system with parameter feedback also maintains stability for higher rate parameter changes than the open loop system. Modeling the systems as affine parameter dependent systems and considering only the unforced case for stability analysis, we have $\mathbf{A}(\theta) \equiv \tilde{\mathbf{A}} + \hat{\mathbf{A}}\theta$ for the open loop system and $\mathbf{A}_{CL}(\theta) \equiv \tilde{\mathbf{A}}_{CL} + \hat{\mathbf{A}}_{CL}\theta$ for the parameter feedback system. Using the quadratic Lyapunov function test (4), the system with parameter feedback was stable up to a maximum parameter variation rate of just over $\dot{\theta} = [-99,99]$. The open loop system, on the other hand, was stable up to a maximum parameter variation rate of just over $\dot{\theta} = [-.5,.5]$. Parameter feedback allows rate variations nearly 200 times faster than the open loop system.

Observer Based Feedback: Observer based feedback can also be used to improve a system's robust stability. Using a similar approach to the state feedback case, the robustness of following closed loop LPV plant and observer controller may be improved using parameter feedback if $\{\mathbf{A}(\theta(t)), \mathbf{B}(\theta(t))\}$ is controllable and $\{\mathbf{A}(\theta(t)), \mathbf{C}(\theta(t))\}$ is observable for all $\theta(t)$ within the desired operating range.

$$\begin{aligned}
&\dot{\mathbf{x}}(t) = \mathbf{A}(\theta(t))\mathbf{x}(t) + \mathbf{B}(\theta(t))\mathbf{u}(t) \\
&\dot{\hat{\mathbf{x}}}(t) = \mathbf{A}(\overline{\theta}(t))\hat{\mathbf{x}}(t) + \mathbf{B}(\overline{\theta}(t))\mathbf{u}(t) + \mathbf{H}(\overline{\theta}(t))(y - \mathbf{C}(\overline{\theta}(t))\hat{\mathbf{x}}(t)) \\
&\mathbf{u}(t) = \mathbf{F}(\overline{\theta}(t))\hat{\mathbf{x}}(t) \\
&\mathbf{y}(t) = \mathbf{C}(\theta(t))\mathbf{x}(t)
\end{aligned}$$

Suitable selection of the regulator and observer gains, $\mathbf{F}(\overline{\theta}(t))$ and $\mathbf{H}(\overline{\theta}(t))$, respectively, will remove the parameter from the closed loop frozen parameter eigenvalues. By the separation principle, these eigenvalues are those of regulator dynamics matrix, $\mathbf{A}(\theta(t)) + \mathbf{B}(\theta(t))\mathbf{F}(\overline{\theta}(t))$, and the observer dynamics matrix, $\mathbf{A}(\overline{\theta}(t)) - \mathbf{H}(\overline{\theta}(t))\mathbf{C}(\overline{\theta}(t))$. As in the state feedback case, the gain, $\mathbf{F}(\overline{\theta}(t))$, is computed by solving the n equations found by comparing the coefficients of the desired regulator's characteristic polynomial with the characteristic polynomial of $\mathbf{A}(\theta(t)) + \mathbf{B}(\theta(t))\mathbf{F}(\overline{\theta}(t))$. Likewise, the observer gain, $\mathbf{H}(\overline{\theta}(t))$, is found from the n equations derived by comparing the coefficients of the desired observer's characteristic polynomial with the characteristic polynomial of $\mathbf{A}(\overline{\theta}(t)) - \mathbf{H}(\overline{\theta}(t))\mathbf{C}(\overline{\theta}(t))$. Similar improvements in robustness were achieved using observer feedback as were obtained with state feedback.

Output Feedback: The generalized parameter varying system (1), could describe a LPV plant and its performance objectives. Gain scheduling is one approach to designing a controller or family of controllers for (1). For example, LTI output feedback controllers could be designed for particular operating points or fixed values of the parameters, θ. The LTI controllers could be developed using H_∞/μ synthesis [1] to provide robust stability and performance when operating about the nominal parameter values θ. Additional robustness may be achieved by augmenting the LTI control with parameter feedback.

Consider an LPV plant, $G(\theta(t))$ and an LTI output feedback controller, $K(s)$, which combine to form the closed loop system, $P_{CL}(\theta(t))$, with the following frozen parameter transfer functions:

$$G(s,\theta)=\frac{b(s,\theta)}{a(s,\theta)}\,,\qquad K(s)=\frac{p(s)}{q(s)}\,,\qquad P_{CL}(s,\theta)=\frac{n(s,\theta)}{d(s,\theta)}=\frac{p(s)b(s,\theta)}{q(s)a(s,\theta)+p(s)b(s,\theta)}\,.$$

By augmenting the LTI controller $K(s)$ with parameter dependent terms, the closed loop characteristic polynomial may be rendered parameter independent. The augmented controller, $K_{\overline{\theta}}(s)$, must be of sufficient order to allow cancellation of the parameters. For example, let $a(s,\theta)$ be of order n, $b(s,\theta)$ be of order m, p be of order j and q be of order k. We restrict ourselves to strictly proper transfer functions such that n>m and k>j as well as controllers of order at least as large as the plant, n≥k. The LTI controller and the parameter augmented controller's transfer function are as follows:

$$K(s)=\frac{p(s)}{q(s)}=\frac{p_j s^j+p_{j-1}s^{j-1}+\cdots+p_0}{s^k+q_{k-1}s^{k-1}+\cdots+q_0}$$

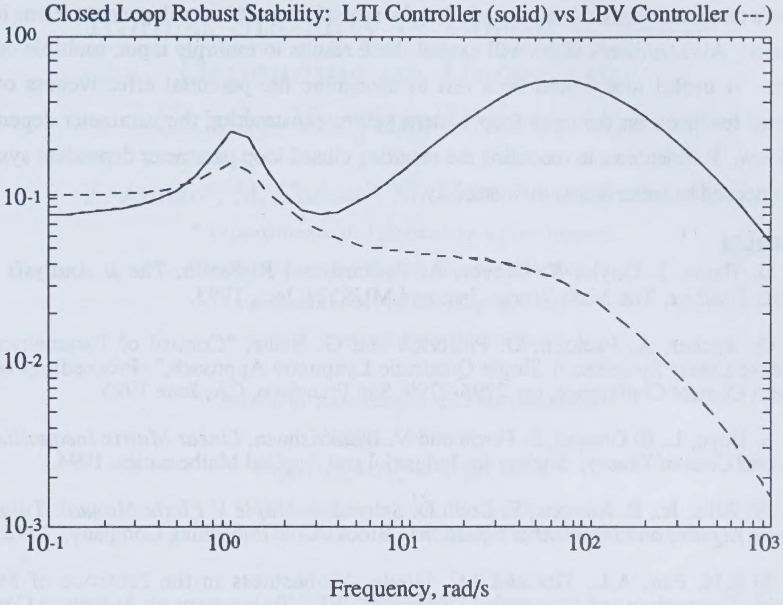
$$K_{\overline{\theta}}(s)=\frac{p_{\overline{\theta}}(s,\overline{\theta})}{q_{\overline{\theta}}(s,\overline{\theta})}=\frac{(p_j+\alpha_j(\overline{\theta}))s^j+(p_{j-1}+\alpha_{j-1}(\overline{\theta}))s^{j-1}+\cdots+(p_0+\alpha_0(\overline{\theta}))}{s^k+(q_{k-1}+\beta_{k-1}(\overline{\theta}))s^{k-1}+\cdots+(q_0+\beta_0(\overline{\theta}))}$$

The controller has a total of (j+k+1) free parameter dependent terms to use. The desired closed loop characteristic equation is of order (n+k). Therefore, comparing the desired closed loop polynomial and $d(s,\theta)$ leads to n+k equations. To solve these equations for the parameter dependent terms, α and β, the total number of required α, β terms is n+k. Depending on the structure of the controller, more parameters may be available then necessary. Selecting n+k parameters α, β and solving for these selected parameters allows for construction of the augmented controller, $K_{\overline{\theta}}(s)$. $K_{\overline{\theta}}(s)$ will cancel the parameter dependent terms from the closed loop characteristic equation when $\overline{\theta}(t)=\theta(t)$. The augmented controller may improve the closed loop system robust stability and performance even with significant parameter measurement errors, $\overline{\theta}(t)\neq\theta(t)$.

Example 2: Refer again the system from Example 1, $P(\theta(t))$. The objective is to design an output feedback regulator so that the output y, tracks a command r with a bandwidth of 1 rad/s. The performance objectives may be represented by a system of LPV equations similar to (1). An LTI controller, $K(s)$, designed to meet the objectives for operation about $\theta=0$ was found after three H_∞/ μ synthesis iterations and subsequent order reduction:

$$K(s)=\left[\begin{array}{cccc|c}
-1.0897e+03 & 1.5591e+03 & 2.0712e+01 & -5.1521e-02 & 3.5017e+02\\
-1.2156e+03 & -2.7488e+01 & 2.1390e+01 & 8.0779e-03 & -5.4547e+01\\
0 & 0 & -3.4707e+02 & 1.7008e-03 & -1.0761e+01\\
0 & 0 & 0 & -1.0000e-05 & 2.6736e-01\\
\hline
3.5009e+02 & -5.5850e+01 & 4.1960e+00 & 2.6735e-01 & 0
\end{array}\right]$$

Then this controller was augmented with parameter dependent terms so that the closed loop eigenvalues were parameter independent when $\overline{\theta}(t)=\theta(t)$. As the μ plots below shows, the closed loop system with the parameter dependent controller has improved robust stability compared the original closed loop system with the LTI controller. Both plots are for complex perturbations or variations in the parameter. The LPV controller's μ plot is for $\overline{\theta}(t) = \theta(t)$.


Closed Loop Robust Stability: LTI Controller (solid) vs LPV Controller (---)

The stability margins for the closed loop systems with regard to strictly real perturbations or parameter variations are $1/\mu_{ub\,LTI} = 1.86$ and $1/\mu_{ub\,LPV} = 1000$, for the LTI and LPV controllers respectively. This shows that the closed loop system with the LPV controller has much greater robustness against time-invariant, real parameter variations than the closed loop system with the LTI controller.

When the parameter varies with time, the LPV controller has improved robustness as compared to the LTI controller. Both cases, modeled as affine parameter dependent systems, were stable using the quadratic Lyapunov function test (4) with $\dot{\theta} = [-\infty, \infty]$. By using a matrix dilation [7], [3], the factor by which the region of stability could be expanded beyond the parameter box, $\theta = [-1, 1]$, can be computed. The LPV controller demonstrated greater robustness than the LTI controller, expanding the stability region by a factor of 2.67 as opposed to 1.84.

Conclusion

As shown with numerical examples, parameter feedback can improve the robust stability and performance as measured by the structured singular value, μ, and parameter dependent Lyapunov functions. The parameter feedback may be used alone or may augment LTI controllers, including state feedback controllers, observer based controllers and output feedback controllers. Future work includes investigating the effect of using all terms of the output feedback controller to not only eliminate the parameters from the closed loop transfer function's denominator but to also minimize the size of the parameter dependent terms in the numerator. Also, further efforts will extend these results to multiple input, multiple output systems. A useful tool would be a test to determine the potential effectiveness of the parameter feedback on the open loop system before constructing the parameter dependent control law. Refinements in modeling the resulting closed loop parameter dependent systems are also needed to better assess their stability.

References

[1] G. Balas, J. Doyle, K. Glover, A. Packard and R. Smith, *The μ Analysis And Synthesis Toolbox*, The Math Works, Inc. and MUSYN, Inc., 1993.

[2] G. Becker, A. Packard, D. Philbrick and G. Balas, "Control of Parametrically-Dependent Linear Systems: A Single Quadratic Lyapunov Approach," Proceedings of the American Control Conference, pp. 2795-2799, San Francisco, CA, June 1993.

[3] S. Boyd, L. El Ghaoui, E. Feron and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory,* Society for Industrial and Applied Mathematics, 1994.

[4] W. Ellis, Jr., E. Johnson, E. Lodi, D. Schwalbe, *Maple V Flight Manual: Tutorials for Linear Algebra and Differential Equations*, Brooks/Cole Publishing Company, 1992.

[5] M.K.H. Fan, A.L. Tits and J.C. Doyle, "Robustness in the Presence of Mixed Parametric Uncertainty and Unmodeled Dynamics," IEEE Transactions on Automatic Control, vol. AC-36, No. 1, pp. 25-38, January 1991.

[6] B. Friedland, J. Richman and D. Williams, "On the "Adiabatic Approximation" for Design of Control Laws for Linear, Time-Varying Systems," IEEE Transactions on Automatic Control, vol. AC-32, No. 1, pp. 62-63, January 1987.

[7] P. Gahinet, A. Nemirovski, A.J. Laub and M. Chilali, *LMI Control Toolbox*, The Mathworks, Inc., 1995.

[8] P. Gahinet, P. Apkarian and M. Chilali, "Affine Parameter-Dependent Lyapunov Functions for Real Parametric Uncertainty," Proceedings of the 33rd Conference on Decision and Control, pp. 2026-2031, Lake Buena Vista, FL, December 1994.

[9] A. Packard and F. Wu, "Control of Linear Fractional Transformations," Proceedings of the 32nd Conference on Decision and Control, pp. 1036-1041, San Antonio, TX, December 1993.

[10] W. Rugh, "Analytical Framework for Gain Scheduling," IEEE Control Systems Magazine, pp. 79-84, January 1991.

[11] J. Shamma and M. Athans, "Guaranteed Properties of Gain Scheduled Control for Linear Parameter-varying Plants," Automatica, vol. 27, No. 3, pp. 559-564, 1991.

[12] J. Shamma and M. Athans, "Gain Scheduling: Potential Hazards and Possible Remedies," IEEE Control Systems Magazine, pp. 101-107, June 1992.

[13] G. Stein, G. Hartmann, and R. Hendrick, "Adaptive Control Laws for F-8 Flight Test," IEEE Transactions on Automatic Control, vol. AC-22, No. 5, pp. 758-767, October 1977.

Robust Stability of Linear Systems Depending on Uncertain, Slowly-Varying Parameters

F. Amato*, M. Corless†, M. Mattei* and R. Setola*

* Dipartimento di Informatica e Sistemistica
Università degli Studi di Napoli Federico II
via Claudio 21, 80125 Napoli, Italy
Email: amato@disna.dis.unina.it

† School of Aeronautics and Astronautics
Purdue University
West Lafayette, IN 47907, USA
Email: corless@ecn.purdue.edu

Abstract

In this paper we consider an uncertain linear system depending affinely on a vector of uncertain parameters. Assuming that the nominal system (which is obtained by setting all parameters to zero) is exponentially stable, exponential stability is maintained in the face of any time-varying realization of parameters whose magnitudes are smaller than the *quadratic stability radius* and for any time-invariant realization of parameters whose magnitudes are smaller than the *multivariable gain margin*. In the presence of time-varying parameters whose magnitudes are between these two stability measures, stability may depend on the rate of variation of the parameters. In this case, as shown in this paper, sufficient conditions for stability can be stated in terms of the existence of suitable parameter dependent Lyapunov functions.

1 Introduction

In the past decade, two major approaches have emerged to investigate the stability of a linear system containing uncertain parameters. The first one assumes that the uncertainties are time-invariant; it establishes whether or not the eigenvalues of the system matrix are in the open left half plane for all allowable values of the uncertainties; see [4] and the bibliography therein. When the uncertainties are time-varying, this approach fails. Indeed *robust stability* with respect to constant parameters (i.e. stability for each fixed value of the uncertain parameters), does not guarantee stability when the parameters are time-varying ([7], [2]).

For time-varying parameters, another approach, called the *quadratic stability* approach can be utilized; see [6] and the bibliography therein. In this approach one looks for a quadratic Lyapunov function which is independent of the uncertainty and which guarantees stability for all allowable values of the uncertainty. Roughly speaking, because the Lyapunov function is independent of the uncertainty, this approach guarantees uniform exponential stability when the parameters are time-varying; also there is no restriction on how fast the parameters can vary.

In this paper, we consider linear systems, depending affinely on a vector of uncertain parameters, which are robustly stable with respect to constant parameters but which are *not* quadratically stable. We assume, as is often the case in real world situations, that the uncertain parameters are time-varying with a known bound and a known *bounded rate.* In this case a condition guaranteeing uniform exponential stability can be found with the aid of *parameter dependent* quadratic Lyapunov functions; this approach allows one to take into account the rate of variation of the parameters. First, in Section 3, we consider the *Lur'e-Postnikov* class (Narendra and Taylor, 1973) which is a strict subset of the family of affinely dependent quadratic Lyapunov functions. We show how to convert the problem of finding a suitable function in this class, that is a function which enables one to prove the stability of the system, to a convex optimization problem. Then, in Section 4, we enlarge the class of admissible Lyapunov functions to the whole family of affine functions; again the problem is converted into a convex optimization algorithm at the price of some conservatism.

2 Problem Statement and Preliminary Results

In this paper, we consider the stability properties of the classical uncertain system (see [11]) described by the following equation

$$\dot{\boldsymbol{x}}(t) = (\boldsymbol{A} + \Delta\boldsymbol{A})\boldsymbol{x}(t) \tag{1}$$

where the certain matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is assumed to be asymptotically stable. The uncertain matrix $\Delta\boldsymbol{A}$ is in the form

$$\Delta\boldsymbol{A} = \sum_{i=1}^{q} p_i \boldsymbol{b}_i \boldsymbol{c}_i^T \tag{2}$$

where $\boldsymbol{b}_i \in \mathbb{R}^n$, $\boldsymbol{c}_i \in \mathbb{R}^n$ are certain and the scalars p_i, are uncertain (possibly time-varying) parameters. Note that system (1) is equivalent to the feedback system described by the state space equations

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} && \text{(3a)}\\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} && \text{(3b)}\\ \boldsymbol{u} &= \boldsymbol{\Pi}\boldsymbol{y}\,. && \text{(3c)} \end{aligned}$$

where $\boldsymbol{B} = (\,\boldsymbol{b}_1 \;\; \ldots \;\; \boldsymbol{b}_q\,)$, $\boldsymbol{C} = (\,\boldsymbol{c}_1 \;\; \ldots \;\; \boldsymbol{c}_q\,)^T$ and $\boldsymbol{\Pi} = \operatorname{diag}(\boldsymbol{p})$ with $\boldsymbol{p} := (\,p_1 \;\; \ldots \;\; p_q\,)^T$.

We can associate two different robustness measures with system (1):

$$\begin{aligned} \rho_Q &:= \inf\{\|\boldsymbol{p}\|_\infty : \boldsymbol{p} \in \mathbb{R}^q \text{ and system (1) is not quadratically stable}\} && (4)\\ \rho_G &:= \inf\{\|\boldsymbol{p}\|_\infty : \boldsymbol{p} \in \mathbb{R}^q \text{ and } \operatorname{sp}(\boldsymbol{A} + \sum_{i=1}^{q} \boldsymbol{b}_i p_i \boldsymbol{c}_i^T) \cap \overline{\mathbb{C}^+} \neq \emptyset\} && (5) \end{aligned}$$

where $\overline{\mathbb{C}^+}$ denotes the closed right half of the complex plane; given a vector $\boldsymbol{v} = (\,v_1 \;\; \ldots \;\; v_q\,)^T$, $\|\boldsymbol{v}\|_\infty := \max\{|v_1|, \ldots, |v_q|\}$; and $\operatorname{sp}(\boldsymbol{F})$ denotes the set of eigenvalues of the square matrix

F. Finally, the statement, "system (1) is quadratically stable", means that there exists a quadratic, time-invariant Lyapunov function $V(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{P}\boldsymbol{x}$ whose derivative along solutions of system (1) is negative definite for all possible values of the uncertain parameters. These two robustness measures are known in the literarature as the *quadratic stability radius* and the *multivariable gain margin* (see [8] and [5]).

The quadratic stability radius has a deeper meaning in the context of robust stability in the presence of *time-varying* perturbations. Indeed system (1) is uniformly exponentially stable for all Lebesgue measurable real time-varying parameters $\boldsymbol{p}(\,\cdot\,)$ satisfying $\|\boldsymbol{p}(\,\cdot\,)\|_\infty := \sup_t \|\boldsymbol{p}(t)\|_\infty < \rho_Q$.

Clearly, $\rho_Q \leq \rho_G$. In general this inequality is strict as many examples illustrate (see for example [6] and [9]).

In this paper, we consider the situation in which the vector $\boldsymbol{p}$ of uncertain parameters is time-varying and satisfies

$$\|\boldsymbol{p}(\,\cdot\,)\|_\infty \leq \overline{p} \tag{6}$$

for some known bound $\overline{p}$. If $\rho_Q = \rho_G$, then robust stability with respect to constant real parameters is equivalent to robust stability with respect to time-varying parameters. So, we consider the case in which

$$\rho_Q < \overline{p} < \rho_G$$

We also suppose that

$$\|\dot{\boldsymbol{p}}(\,\cdot\,)\|_\infty \leq \overline{h} \tag{7}$$

where $\overline{h}$ is known. We will let $\mathcal{P}_{\overline{p},\overline{h}}$ be the class of differentiable real vector-valued functions $\boldsymbol{p}(\cdot)$ defined on $[0,+\infty)$ and satisfying bounds (6) and (7).

First we have the following observations.

i) Exponential stability is not guaranteed for arbitrary time-varying $\boldsymbol{p}(\,\cdot\,)$ because $\overline{p} > \rho_Q$, and the system *is not* quadratically stable.

ii) Exponential stability is guaranteed for constant $\boldsymbol{p}$ because $\overline{p} < \rho_G$.

iii) For a given $\overline{p}$, exponential stability is guaranteed for $\overline{h}$ sufficiently small, i.e., for time-varying $\boldsymbol{p}(\,\cdot\,)$ which are slowly varying in time ([7], [2]).

We consider the following two problems.

Problem 1 Given system (1) with $\boldsymbol{p}(\,\cdot\,)$ being a vector of *unknown* time-varying parameters belonging to the class of functions $\mathcal{P}_{\overline{p},\overline{h}}$, where $\rho_Q < \overline{p} < \rho_G$, *establish* a condition for the uniform exponential stability of system (1) for all possible $\boldsymbol{p}(\,\cdot\,) \in \mathcal{P}_{\overline{p},\overline{h}}$.

Problem 2 Given $\rho_Q < \overline{p} < \rho_G$, *find* an estimate of the supremal $\overline{h}$ such that system (1) is uniformly exponentially stable for all $\boldsymbol{p}(\,\cdot\,) \in \mathcal{P}_{\overline{p},\overline{h}}$.

Our effort will be devoted to solving the first problem; the latter problem can be reduced to a simple binary search algorithm based on the solution of Problem 1.

To take into account the rate of variation of $\boldsymbol{p}(\cdot)$ in the study of the stability properties of system (1), we need to use a Lyapunov function depending on $\boldsymbol{p}$. This will be given by $\boldsymbol{x}^T \boldsymbol{P}(\boldsymbol{p})\boldsymbol{x}$ where $\boldsymbol{P}(\cdot)$ is a differentiable, symmetric, matrix-valued function which satisfies the following condition. We denote by $\boldsymbol{h}^{(j)}$, $j = 1, \ldots, 2^q$, the vertices of the hyperrectangle $\mathcal{R}_{\overline{h}}$ in $\mathbb{R}^q$ defined by $\|\boldsymbol{h}\|_\infty \leq \overline{h}$, that is

$$\mathcal{R}_{\overline{h}} := [-\overline{h},\overline{h}] \times [-\overline{h},\overline{h}] \times \cdots \times [-\overline{h},\overline{h}] = [-\overline{h},\overline{h}]^q \subset \mathbb{R}^q \tag{8}$$

Condition 1 *For* $\|p\|_\infty \le \bar{p}$,

$$P(p) > 0 \tag{9}$$

and

$$(A+\sum_{i=1}^{q} b_i p_i c_i^T)^T P(p)+P(p)(A+\sum_{i=1}^{q} b_i p_i c_i^T)+\sum_{i=1}^{q} h_i^{(j)} P_i'(p) < 0 \qquad for \quad j = 1,\ldots,2^q \tag{10}$$

where $P_i'(\cdot)$ *denotes the partial derivative of* $P(\cdot)$ *with respect to* p_i.

We have now the following theorem.

Theorem 1 *Suppose there exists a differentiable, symmetric matrix-valued function* $P(\cdot)$ *which satisfies Condition 1. Then system (1) is uniformly exponentially stable for all* $p(\cdot) \in \mathcal{P}_{\bar{p},\bar{h}}$.

Proof. Let $p(\cdot)$ be any realization of the uncertain parameter in the set $\mathcal{P}_{\bar{p},\bar{h}}$. Consider the time-varying system

$$\dot{x}(t) = (A + \sum_{i=1}^{q} b_i p_i(t) c_i^T) x(t) \tag{11}$$

and the candidate Lyapunov function given by $V(t,x) := x^T P(p(t)) x$. First note that

$$\lambda_{\min}(P(p))||x||^2 \ \le \ x^T P(p) x \ \le \ \lambda_{\max}(P(p))||x||^2$$

where $\lambda_{\min}(P(p))$ and $\lambda_{\max}(P(p))$ denote the minumum and maximum eigenvalues of $P(p)$, respectively. Since the minimum and maximun eigenvalues of a symmetric matrix depend continuously on the matrix, $P(\cdot)$ is continuous and p ranges in a compact set, it follows from (9) that there are positive constants ω_1 and ω_2 such that for all p with $||p||_\infty \le \bar{p}$,

$$0 < \omega_1 \le \lambda_{\min}(P(p)); \qquad \lambda_{\max}(P(p)) \le \omega_2$$

Hence,

$$\omega_1||x||^2 \le V(t,x) \le \omega_2||x||^2 \tag{12}$$

The derivative of V along the trajectories of system (11) is given by

$$\dot{V}(t,x) = -x^T Q(p(t),\dot{p}(t)) x \tag{13}$$

where

$$-Q(p,\dot{p}) := (A + \sum_{i=1}^{q} b_i p_i c_i^T)^T P(p) + P(p)(A + \sum_{i=1}^{q} b_i p_i c_i^T) + \sum_{i=1}^{q} \dot{p}_i P_i'(p).$$

Since $Q(p,h)$ depends affinely on h, it follows that condition (10) is equivalent to

$$-\,Q(p,h) < 0 \qquad \text{for} \quad ||h||_\infty \le \bar{h} \tag{14}$$

Since $P(\cdot)$ is continuous and (p,h) ranges in a compact set, it follows that there exists a positive constant ω_3 such that for all p, h with $||p||_\infty \le \bar{p}$ and $||h||_\infty \le \bar{h}$,

$$0 < \omega_3 \le \lambda_{\min}(Q(p,h))$$

Hence

$$\dot{V}(t,x) \le -\omega_3||x||^2 \tag{15}$$

Conditions (12) and (15) guarantee that system (1) is globally uniformly exponentially stable for all $p(\cdot) \in \mathcal{P}_{\bar{p},\bar{h}}$. Also the rate of convergence is independent of the uncertain parameter. ■

In the next section we discuss how to find a suitable matrix-valued function $P(\cdot)$.

A Sufficient Condition for Exponential Stability

The possibility of solving Problem 1 in a simple computational way strongly depends on the structure of the matrix-valued function $\boldsymbol{P}(\cdot)$. In this section we consider functions $\boldsymbol{P}(\cdot)$ which lead to a very simple sufficient condition guaranteeing Condition 1; more precisely we search for a parameter-dependent Lyapunov matrix in the form of Lur'è Postnikov (see [11] and the bibliography therein). To this end, let

$$\mathcal{L} := \left\{ \boldsymbol{P}(\cdot) \mid \boldsymbol{P}(\boldsymbol{p}) = \boldsymbol{R} + \sum_{i=1}^{q} \mu_i p_i \boldsymbol{c}_i \boldsymbol{c}_i^T,\ \boldsymbol{R}^T = \boldsymbol{R},\ \mu_i \in \mathbb{R},\ i = 1, \ldots, q \right\} \tag{16}$$

Now we state the main result of this section, which is a sufficient condition for the existence of a $\boldsymbol{P}(\cdot) \in \mathcal{L}$ satisfying Condition 1. To this end let us denote by $\boldsymbol{p}^{(j)}$, $j = 1, \ldots, 2^q$, the vertices of the hyper-rectangle $\mathcal{R}_{\overline{p}}$ in $\mathbb{R}^q$ defined by $\|\boldsymbol{p}\|_\infty \leq \overline{p}$, that is

$$\mathcal{R}_{\overline{p}} := [-\overline{p}, \overline{p}]^q \subset \mathbb{R}^q \tag{17}$$

Theorem 2 *There exists a symmetric matrix valued function $\boldsymbol{P}(\cdot)$ belonging to class $\mathcal{L}$ and satisfying Condition 1* if *there exist a real symmetric matrix $\boldsymbol{R}$ and real scalars μ_i, τ_{ij}, $i = 1, \ldots, q$, $j = 1, \ldots, 2^q$, with*

$$\tau_{ij} \geq 0 \tag{18}$$

such that for $j = 1, \ldots, 2^q$,

$$\boldsymbol{R} + \sum_{i=1}^{q} \mu_i p_i^{(j)} \boldsymbol{c}_i \boldsymbol{c}_i^T > 0 \tag{19}$$

$$\begin{pmatrix} \boldsymbol{A}^T\boldsymbol{R} + \boldsymbol{R}\boldsymbol{A} + \sum_{i=1}^{q} \mu_i \boldsymbol{c}_i \boldsymbol{c}_i^T h_i^{(j)} + \overline{p}^2 \sum_{i=1}^{q} \tau_{ij} \boldsymbol{c}_i \boldsymbol{c}_i^T & \mu_1 \boldsymbol{A}^T \boldsymbol{c}_1 + \boldsymbol{R}\boldsymbol{b}_1 & \ldots & \mu_q \boldsymbol{A}^T \boldsymbol{c}_q + \boldsymbol{R}\boldsymbol{b}_q \\ \boldsymbol{b}_1^T \boldsymbol{R} + \mu_1 \boldsymbol{c}_1^T \boldsymbol{A} & \mu_1(\boldsymbol{b}_1^T \boldsymbol{c}_1 + \boldsymbol{c}_1^T \boldsymbol{b}_1) - \tau_{1j} & \ldots & \mu_q \boldsymbol{b}_1^T \boldsymbol{c}_q + \mu_1 \boldsymbol{c}_1^T \boldsymbol{b}_q \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{b}_q^T \boldsymbol{R} + \mu_q \boldsymbol{c}_q^T \boldsymbol{A} & \mu_1 \boldsymbol{b}_q^T \boldsymbol{c}_1 + \mu_q \boldsymbol{c}_q^T \boldsymbol{b}_1 & \ldots & \mu_q(\boldsymbol{b}_q^T \boldsymbol{c}_q + \boldsymbol{c}_q^T \boldsymbol{b}_q) - \tau_{qj} \end{pmatrix} < 0 \tag{20}$$

Proof. We first note that conditions (19) hold if and only if condition (9) holds for all $\boldsymbol{p}$ with $\|\boldsymbol{p}\|_\infty \leq \overline{p}$.

Letting $u_i = p_i \boldsymbol{c}_i^T \boldsymbol{x}$ for $i = 1, \ldots, q$, $\boldsymbol{u} = (u_1 \quad u_2 \quad \ldots \quad u_q)^T$ and considering any $j = 1, \ldots, 2^q$, we have

$$\begin{aligned}
&\boldsymbol{x}^T \left[(\boldsymbol{A} + \sum_{i=1}^{q} \boldsymbol{b}_i p_i \boldsymbol{c}_i^T)^T (\boldsymbol{R} + \sum_{i=1}^{q} \mu_i p_i \boldsymbol{c}_i \boldsymbol{c}_i^T) + (\boldsymbol{R} + \sum_{i=1}^{q} \mu_i p_i \boldsymbol{c}_i \boldsymbol{c}_i^T)(\boldsymbol{A} + \sum_{i=1}^{q} \boldsymbol{b}_i p_i \boldsymbol{c}_i^T) + \sum_{i=1}^{q} \mu_i h_i^{(j)} \boldsymbol{c}_i \boldsymbol{c}_i^T \right] \boldsymbol{x} \\
&\quad = \boldsymbol{x}^T (\boldsymbol{A}^T \boldsymbol{R} + \boldsymbol{R}\boldsymbol{A} + \sum_{i=1}^{q} \mu_i h_i^{(j)} \boldsymbol{c}_i \boldsymbol{c}_i^T) \boldsymbol{x} \\
&\quad + \sum_{i=1}^{q} \boldsymbol{x}^T (\boldsymbol{R}\boldsymbol{b}_i) u_i + \sum_{i=1}^{q} u_i (\boldsymbol{b}_i^T \boldsymbol{R}) \boldsymbol{x} + \sum_{i=1}^{q} u_i (\mu_i \boldsymbol{c}_i^T \boldsymbol{A}) \boldsymbol{x} + \sum_{i=1}^{q} \boldsymbol{x}^T (\mu_i \boldsymbol{A}^T \boldsymbol{c}_i) u_i \\
&\quad + \sum_{i=1}^{q} \sum_{j=1}^{q} u_i (\mu_i \boldsymbol{c}_i^T \boldsymbol{b}_j) u_j + \sum_{i=1}^{q} \sum_{j=1}^{q} u_j (\mu_i \boldsymbol{b}_j^T \boldsymbol{c}_i) u_i \\
&\quad = \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{u} \end{pmatrix}^T \boldsymbol{S}_j \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{u} \end{pmatrix}
\end{aligned}$$

where

$$\boldsymbol{S}_j := \begin{pmatrix} \boldsymbol{A}^T\boldsymbol{R} + \boldsymbol{R}\boldsymbol{A} + \sum_{i=1}^q \mu_i \boldsymbol{c}_i\boldsymbol{c}_i^T h_i^{(j)} & \mu_1 \boldsymbol{A}^T\boldsymbol{c}_1 + \boldsymbol{R}\boldsymbol{b}_1 & \dots & \mu_q \boldsymbol{A}^T\boldsymbol{c}_q + \boldsymbol{R}\boldsymbol{b}_q \\ \boldsymbol{b}_1^T\boldsymbol{R} + \mu_1\boldsymbol{c}_1^T\boldsymbol{A} & \mu_1(\boldsymbol{b}_1^T\boldsymbol{c}_1 + \boldsymbol{c}_1^T\boldsymbol{b}_1) & \dots & \mu_q\boldsymbol{b}_1^T\boldsymbol{c}_q + \mu_1\boldsymbol{c}_1^T\boldsymbol{b}_q \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{b}_q^T\boldsymbol{R} + \mu_q\boldsymbol{c}_q^T\boldsymbol{A} & \mu_1\boldsymbol{b}_q^T\boldsymbol{c}_1 + \mu_q\boldsymbol{c}_q^T\boldsymbol{b}_1 & \dots & \mu_q(\boldsymbol{b}_q^T\boldsymbol{c}_q + \boldsymbol{c}_q^T\boldsymbol{b}_q) \end{pmatrix}$$

Hence condition (10) holds for $\|p\|_\infty \leq \overline{p}$ if and only if the following holds for $j = 1, \dots, 2^q$:

$$\begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{u} \end{pmatrix}^T \boldsymbol{S}_j \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{u} \end{pmatrix} < 0 \quad \text{whenever} \quad x \neq 0 \quad \text{and} \quad \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{u} \end{pmatrix}^T \boldsymbol{M}_i \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{u} \end{pmatrix} \leq 0 \quad \text{for} \quad i = 1, \dots, q \tag{21}$$

where $\boldsymbol{M}_i$ is the block diagonal matrix

$$\boldsymbol{M}_i = \begin{pmatrix} -\overline{p}^2\boldsymbol{c}_i\boldsymbol{c}_i^T & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{D}_i \end{pmatrix} \tag{22}$$

and

$$(\boldsymbol{D}_i)_{kl} = \begin{cases} 1 & \text{if } k = l = i \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

It can readily be shown that condition (21) is satisfied *if* there exist scalars $\tau_{ij} \geq 0$, $i = 1, \dots, q$, such that

$$\boldsymbol{S}_j - \sum_{i=1}^q \tau_{ij}\boldsymbol{M}_i < 0 \tag{24}$$

From this the desired result follows. ■

We conclude that a function $\boldsymbol{P}(\cdot)$ in the class $\mathcal{L}$ satisfying Condition 1 exists *if* the following problem admits a solution.

Problem 3 Find a real symmetric matrix $\boldsymbol{R}$ and real scalars μ_i, τ_{ij}, $i = 1, \dots, q$, $j = 1, \dots, 2^q$, which satisfy conditions (18)-(20).

Problem 3, and hence Problem 1, is a classical convex optimization problem involving LMIs (linear matrix inequalities) and can be solved by one of the algorithms proposed in [5].

Example 1 Consider system (1) with

$$\boldsymbol{A} = \begin{pmatrix} 0 & 1 \\ -2 & -1 \end{pmatrix}, \quad \boldsymbol{b}_1 = \boldsymbol{b}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \boldsymbol{c}_1^T = (\,1 \quad 0\,), \quad \boldsymbol{c}_2^T = (\,0 \quad 1\,) \tag{25}$$

In this case $\rho_Q = 0.583$ and $\rho_G = 1$. Recalling Problem 2, we have computed the maximum allowable rate of variation of the parameters for some values of $\overline{p}$ between 0.6 and 1. The results are presented in the following table.

$\overline{p}$	0.6	0.7	0.8	0.9
$\overline{h}_{max}$	1.25	0.77	0.46	0.21

In particular with $\overline{p} = 0.6$, $\overline{h} = 1.25$ a candidate Lyapunov function has been found with

$$\begin{aligned} \boldsymbol{R} &= \begin{pmatrix} 0.8321 & 0.1311 \\ 0.1311 & 0.4170 \end{pmatrix}; \quad \mu_1 = -0.2429; \quad \mu_2 = -2.437 \cdot 10^{-4}; \\ \tau_{11} &= 0.4735; \quad \tau_{12} = 0.4311; \quad \tau_{13} = 0.4681; \quad \tau_{14} = 0.3303; \\ \tau_{21} &= 0.7571; \quad \tau_{22} = 0.6865; \quad \tau_{23} = 0.7089; \quad \tau_{24} = 0.6824. \end{aligned}$$

A More General Class of Parameter Dependent Lyapunov Functions

The advantage of the procedure detailed in the previous section relies on the fact that conditions (9) and (10) have been converted, in theorem 2, into a finite number of pointwise constraints. The weak points are as follows.

i) Theorem 2 yields only a sufficient condition for the existence of a suitable Lyapunov matrix in the class $\mathcal{L}$. The technical reason for this is that (24) implies (21) but the converse in not true except in the single parameter case. Indeed, when $q = 1$, theorem 2 yields a necessary and sufficient condition (see also [1]).

ii) The class $\mathcal{L}$, when searching for a Lyapunov matrix, is a *strict* subset of the class of affine functions. It is expected that enlarging this class should lead to better bounds than those contained in in the preceding table.

Regarding point ii) above, consider parameter-dependent Lyapunov matrices in the class of affine, symmetric matrix-valued functions

$$\mathcal{A} := \left\{ \boldsymbol{P}(\cdot) \,|\, \boldsymbol{P}(\boldsymbol{p}) = \boldsymbol{R} + \sum_{i=1}^{q} p_i \boldsymbol{R}_i,\ \boldsymbol{R} = \boldsymbol{R}^T,\ \boldsymbol{R}_i = \boldsymbol{R}_i^T,\ i = 1, \ldots, q \right\} \tag{26}$$

The proof of the following result is straightforward.

Theorem 3 *There exists a symmetric matrix valued function* $\boldsymbol{P}(\cdot)$ *belonging to class* $\mathcal{A}$ *and satisfying Condition 1* if and only if *there exist real symmetric matrices* $\boldsymbol{R}$, $\boldsymbol{R}_i$, $i = 1, \ldots, q$, *such that for* $j = 1, \ldots, 2^q$,

$$\boldsymbol{R} + \sum_{i=1}^{q} p_i^{(j)} \boldsymbol{R}_i > 0 \tag{27}$$

and

$$\boldsymbol{A}^T \boldsymbol{R} + \boldsymbol{R}\boldsymbol{A} + \sum_{i=1}^{q} p_i \left[\boldsymbol{R} \boldsymbol{b}_i \boldsymbol{c}_i^T + \boldsymbol{c}_i \boldsymbol{b}_i^T \boldsymbol{R} + \boldsymbol{R}_i \boldsymbol{A} + \boldsymbol{A}^T \boldsymbol{R}_i \right] + \sum_{i=1}^{q} \sum_{k=1}^{q} p_i p_k \left[\boldsymbol{R}_i \boldsymbol{b}_k \boldsymbol{c}_k^T + \boldsymbol{c}_k \boldsymbol{b}_k^T \boldsymbol{R}_i \right] + \sum_{i=1}^{q} h_i^{(j)} \boldsymbol{R}_i < 0 \tag{28}$$

for all $\boldsymbol{p} \in \mathcal{R}_{\overline{p}}$.

Condition (28) cannot be converted into a finite number of pointwise constraints since it contains quadratic terms of the form p_i^2, $i = 1, \ldots, q$.

Now we propose a methodology which, at the price of a heavier computational burden, allows us to convert condition (28) into a finite number of constraints. This procedure introduces some conservatism which can be arbitrarily reduced by increasing the computational effort.

First we provide a procedure which, for the quadratic function $f(x) = x^2$, with $x \in [\underline{x}, \overline{x}]$, yields a multiaffine function $f^m(x, \delta)$, with $\delta \in [0, 1]$, such that

$$f^m([\underline{x}, \overline{x}] \times [0, 1]) \supseteq f([\underline{x}, \overline{x}]). \tag{29}$$

Procedure 1 . Suppose $0 \leq \underline{x} \leq \overline{x}$; let $\overline{f}(\cdot)$ be the affine function connecting the extreme points of the graph of x^2, i.e.

$$\overline{f}(x) = (\overline{x} + \underline{x})x - \underline{x}\overline{x} \tag{30}$$

and let $\underline{f}(\cdot)$ the tangent to the graph of x^2 for $x = \underline{x}$, i.e.

$$\underline{f}(x) := f(\underline{x}) + 2\underline{x}(x - \underline{x}) \tag{31}$$

Then set

$$f^m(x, \delta) = \delta \underline{f}(x) + (1 - \delta)\overline{f}(x) \tag{32}$$

Clearly, (29) holds.

Procedure 1 is a technique for covering the image of the quadratic function $f(x) = x^2$ by that of the multiaffine one f^m and is based on the more general methodology developed in [3]. The dual procedure has to be considered when $\underline{x}$ and $\overline{x}$ are negative numbers.

Since f^m is multiaffine in x and δ, it is positive if and only if it is positive on the four vertices of the rectangle $[\underline{x}, \overline{x}] \times [0, 1]$; moreover the values assumed by f^m on the vertices are

$$\begin{aligned} f^m(\underline{x}, 0) &= \underline{x}^2 && \text{(33a)} \\ f^m(\underline{x}, 1) &= \underline{x}^2 && \text{(33b)} \\ f^m(\overline{x}, 0) &= \underline{x}^2 + 2\underline{x}(\overline{x} - \underline{x}) && \text{(33c)} \\ f^m(\overline{x}, 1) &= \overline{x}^2 && \text{(33d)} \end{aligned}$$

Equations (33) show that we just need to check positiveness on the three vertices $(\underline{x}, 1)$, $(\overline{x}, 0)$ and $(\overline{x}, 1)$; moreover two of the values assumed by f^m on the vertices ((33b) and (33d)) belong also to the image of the original function f ($\underline{x}^2$ and $\overline{x}^2$), while only (33c) is introduced by the new function f^m.

Procedure 2 . Consider the i-th parameter. Select n points (n odd) in the interval $[-\overline{p}, \overline{p}]$, say $\hat{p}_{i,1}$, $\hat{p}_{i,2}$, $\ldots$, $\hat{p}_{i,n}$ with $\hat{p}_{i,1} \equiv -\overline{p}$, $\hat{p}_{i,(n+1)/2} \equiv 0$ and $\hat{p}_{i,n} \equiv \overline{p}$, and apply Procedure 1 to each sub-interval $[\hat{p}_{i,j}, \hat{p}_{i,j+1}]$, $j = 1, \ldots, n-1$. In this way, with $f_i(p_i) = p_i^2$, we define $\underline{f}_i^j$, $\overline{f}_i^j$ such that $f_i^{m,j}(p_i, \delta_{i,j}) := \delta_{i,j}\underline{f}_i^j(p_i) + (1 - \delta_{i,j})\overline{f}_i^j(p_i)$, $j = 1, \ldots, n-1$, $p_i \in [\hat{p}_{i,j}, \hat{p}_{i,j+1}]$ and $\delta_{i,j} \in [0, 1]$.

Following the same considerations above, given $f_i^{m,j}(p_i, \delta_{i,j})$ it is necessary and sufficient to check positiveness of this function at three of the four vertices of $[\hat{p}_{i,j}, \hat{p}_{i,j+1}] \times [0, 1]$. Since $f_i^{m,j}$ and $f_i^{m,j+1}$ have two vertices in common, the total number of checks along the whole interval $[-\overline{p}, \overline{p}]$ is given by $2(n-1) + 1 = 2n - 1$; n of this checks are genuine, that is they involve the true function, while the remaining $n - 1$ introduce conservatism.

Now, using Procedures 1 and 2, we transform (28) into a finite number of constraints as follows.

Procedure 3 Divide into n parts (n odd) the interval $[-\overline{p}, \overline{p}]$ and apply Procedure 2 to each parameter p_i. In this way the original hyper-rectangle $\mathcal{R}_{\overline{p}}$ is divided into n^q sub-hyper-rectangles; we denote by $\mathcal{R}_{j_1, j_2, \ldots, j_q}$ the sub-hyper-rectangle obtained considering the sub-interval $[\hat{p}_{1,j_1}, \hat{p}_{1,j_1+1}]$ for the parameter p_1, $[\hat{p}_{2,j_2}, \hat{p}_{2,j_2+1}]$ for the parameter p_2, etc. Now consider (28) restricted to $\mathcal{R}_{j_1, j_2, \ldots, j_q}$ and replace each term p_i^2 with the corresponding $f_i^{m,j_i}(p_i, \delta_{i,j_i})$. It is clear that the matrix-valued function obtained in this way is multiaffine in the parameters $(p_1, \ldots, p_q, \delta_{1,j_1}, \ldots, \delta_{q,j_q}) \in \mathcal{R}_{j_1,j_2,\ldots,j_q} \times [0,1]^q$. Therefore it is negative definite in $\mathcal{R}_{j_1,j_2,\ldots,j_q} \times [0,1]^q$ *if and only if* it is negative definite on the vertices. According to the discussion above the total number of effective constraints is $(2n-1)^q$ with n^q involving the original function, which we refer to as **necessary constraints**, and the remaining constraints introducing conservatism.

Now we detail the main algorithm we propose to solve Problem 1 with affine Lyapunov functions belonging to the class $\mathcal{A}$.

Algorithm 1

Step 0 Let $n = 3$;

Step 1 Apply Procedure 3. Then attempt to solve the following LMI problem

> **Problem 4** Find symmetric matrices $\boldsymbol{R}$, $\boldsymbol{R}_i$, $i = 1, \ldots, q$, such that (27) is satisfied together with the necessary constraints.
>
> **If** Problem 4 does not admit a feasible solution **then stop** (Problem 1 is unsolvable with the elements of $\mathcal{A}$); **else** try to solve the following problem.
>
> **Problem 5** Find symmetric matrices $\boldsymbol{R}$, $\boldsymbol{R}_i$, $i = 1, \ldots, q$, such that (27) is satisfied together with all the constraints which result from Procedure 3.
>
> **If** Problem 5 admits a feasible solution **then stop** (we have solved Problem 1 with a member of $\mathcal{A}$).

Step 2 Let $n = n + 2$ and go to Step 1.

From a theoretical point of view it is possible to prove that, generically, Algorithm 1 converges in a finite number of iterations. More precisely, if there exists a member of $\mathcal{A}$ which satisfies Condition 1, then there exists an iteration index such that Problem 5 admits a feasible solution; conversely, if there does not exist a member of $\mathcal{A}$ which satisfies Condition 1, then there exists an iteration index such that Problem 4 does not admit a feasible solution. However, one must take into account the practical computational limitations in using Algorithm 1 which arise for large n.

Example 2 Consider again the system in example 1. We have solved Problem 2 applying Algorithm 1 and stopping the procedure when $n = 7$. The following table contains some results. Clearly, these results are less conservative than those obtained previously with the Lur'e Postnikov functions.

$\overline{p}$	0.6	0.7	0.8	0.9
$\overline{h}_{max}$	2.48	1.45	0.79	0.46

In particular with $\overline{p} = 0.6$ and $\overline{h} = 2.5$ a candidate Lyapunov function has been found with

$$\boldsymbol{R} = \begin{pmatrix} 1152.9 & 109.7 \\ 109.7 & 584.8 \end{pmatrix} \quad \boldsymbol{R}_1 = \begin{pmatrix} -94.3 & 13.2 \\ 13.2 & 52.0 \end{pmatrix} \quad \boldsymbol{R}_2 = \begin{pmatrix} 1.33 & -1.25 \\ -1.25 & -2.42 \end{pmatrix}$$

Conclusions

In this paper we have considered an uncertain linear system which depends affinely on several uncertain parameters. The magnitude of each parameter is between the quadratic stability radius and the multivariable gain margin, hence robust stability with respect to constant parameters is ensured but quadratic stability is not. Using a bound on the time derivative of the parameters, we obtain sufficient conditions for the exponential stability of the system with the aid of Lyapunov functions which depend affinely on the parameters.

Acknowledgements

Some of this research was supported by the US National Science Foundation under Grant MSS-90-57079.

References

[1] Amato F. *Stability Analysis for Linear Systems Depending on Uncertain and Time-Varying Parameters.* Phd Thesis, Università di Napoli, Feb. 1994.

[2] Amato F., G. Celentano and F. Garofalo. New sufficient conditions for the stability of slowly varying linear systems. *IEEE Trans. Auto. Contr.*, vol. AC-38, pp. 1409–1411, 1993.

[3] Amato F., F. Garofalo and L. Glielmo. Polytopic coverings and robust stability analysis via Lyapunov quadratic forms. In *Variable Structure and Lyapunov Control*, A. S. I. Zinober (editor), Springer Verlag, Berlin, 1993.

[4] Barmish, B.R.. *New Tools for Robustness Analysis*, 1994. MacMillan.

[5] Boyd S., L. El Ghaoui, E. Feron and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory.* SIAM Press, 1994.

[6] Corless M.. Robust stability analysis and controller design with quadratic Lyapunov Functions. In *Variable Structure and Lyapunov Control*, A. S. I. Zinober (editor), Springer Verlag, Berlin, 1993.

[7] Desoer C. A.. Slowly varying system $\dot{x} = A(t)x$. *IEEE Trans. Auto. Contr.*, vol. AC-14, pp. 780–781, 1969.

[8] De Gaston R. E. and M. G. Safonov. Exact calculation of the multivariable stability margin. *IEEE Trans. Aut. Contr.*, vol. AC-33, pp. 156–171, 1988.

[9] Hinrichsen D. and A. J. Pritchard. Stability radius for structured perturbations and the algebraic Riccati equation. *Syst. Contr. Lett.*, vol. 8, pp. 105–113, 1987.

[10] Khargonekar P. P., I. R. Petersen and K. Zhou. Robust stabilization of uncertain linear systems: quadratic stabilizability and $\mathcal{H}_\infty$ control theory. *IEEE Trans. Auto. Contr.*, vol. AC–35, pp. 356–361, 1990.

[11] Narendra K. S. and J. H. Taylor. *Frequency Domain Criteria for Absolute Stability.* Academic Press, New York, 1973.

A NEW STRATEGY FOR RECONFIGURABLE CONTROL SYSTEMS

Ioannis K. Konstantopoulos and Panos J. Antsaklis
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556, U.S.A.
antsakli@saturn.ee.nd.edu
ikonstan@elgreco.helios.nd.edu

Abstract

An optimization strategy for the problem of control reconfiguration in response to operating condition changes or abrupt system component failures is presented here. The proposed approach provides an output feedback controller that not only stabilizes the new/impaired system, when possible, but also preserves much of the dynamics of the original system. The design is such that the closed-loop system is robust with respect to inevitable uncertainties/modelling errors in the state-space matrices of the impaired system. The approach is applied to an aircraft longitudinal control system, for which two severe cases of failure are considered, first the loss of an actuator and then the loss of a sensor in addition to the actuator loss.

1 Motivation-Previous work

Reconfigurable control systems are control systems that are characterized by the ability to perform in the presence of drastic changes in the system dynamics due, for example, to abrupt system component (actuator/sensor) failures or rapid changes in the operating conditions. Their task is twofold; first they need to guarantee safe performance (stability), when possible, and then recover maximum control performance under impairments. Established techniques exist for the cases of anticipated failures/operating condition changes, for which control laws are precomputed, stored and used according to need. However, the interest here is mainly for the cases of unanticipated scenarios, where an automated on-line failure accommodation technique is needed. Here, we are primarily interested in the reconfiguration part. That is, for the cases of component failures, a failure detection and identification scheme is assumed to provide the dynamics (state-space description) of the impaired system; for the cases of operating condition changes, an on-line modelling technique is assumed to identify the state-space model that corresponds to the new operating conditions. Once the model of the new/impaired system is available, we present a reconfiguration technique to maintain its stability and performance.

Several approaches for aircraft flight control problems have appeared in the literature, [9], [13], [14], [15]. Note also that an interesting overview of the reconfiguration problem for aircraft flight control systems can be found in [6]. Alternative approaches to reconfigurable control are presented in [2], [4], [19], [20]. In all the papers above, reconfiguration is either just a part of a more general adaptive or FDI/stability robustness scheme or treated as an uncertainty that enters the system or restricted to some specific classes of failures for which a control law may be stored and used upon need. However, the requirement of maintaining the dynamics of the original closed-loop system is not explicitly included in the reconfiguration design procedure. This is the way reconfiguration is treated in the papers that follow.

In [3], [16], [17], the *Pseudo-inverse method (PIM)* is used to compute the control law for the impaired system. This method relies on the fact that the new feedback gain based on the pseudo-inverse theory is optimal in the sense that it is the solution of smallest norm for the linear least-squares problem of minimizing the Frobenius norm of the difference matrix between the original and the impaired closed-loop system transition matrices. The main problem, however, is that the stability of the impaired closed-loop system can not be guaranteed. This problem was overcome in [7], where the reconfigurable control problem was formulated as a constrained minimization problem and a *modified pseudo-inverse method (MPIM)* was proposed that guarantees the stability of the closed-loop system. This approach, however, has two drawbacks. First it assumes full state-feedback, which can be quite unrealistic at times; and it relies on some stability bounds that may give very conservative results; this results in certain limitations of the proposed scheme. More recently, an approach was presented in [10], where the full-state measurability (state-feedback) is relaxed and the output feedback case is considered. There are, however, two major drawbacks in the proposed technique. First, the stability of only $Max(m, r)$ eigenvalues of the closed-loop system is maintained, where (m, r) are the numbers of inputs, outputs respectively. Note that similar limitations are encountered in [8]. The second drawback concerns the fact that the proposed methodology relies on the assumption that the input matrix B_f of the impaired model is of full column rank. This is a restrictive assumption, considering the common case of actuator loss which corresponds to zeroing a whole column of the input matrix.

Here, we consider the output feedback case and propose an optimization strategy which guarantees the stability of all the closed-loop eigenvalues, even in the case of severe failure scenarions, such as the simultaneous loss of an actuator and a sensor. Note that this happens under the assumption that a stabilizing solution exists for the impaired state-space model. The new stabilizing feedback controller for the impaired system captures as much of the dynamics of the original system as possible, since it is designed to minimize the Frobenius norm of the difference matrix between the original and the impaired closed-loop transition matrices. Another useful feature of the proposed design is that it is robust with respect to modelling errors in the state-space matrices of the new/impaired system. In other words, the realistic possibility of imperfect modelling of the impaired system is incorporated in our design and the controller derived by the proposed algorithm is capable of maintaining the closed-loop stability even in the presence of uncertainty in the state-space matrices of the impaired system.

2 Problem Formulation

We consider the linear multivariable continuous system with the state-space description

$$\dot{x}(t) = A\,x(t) + B\,u(t), \qquad y(t) = C\,x(t) \tag{1}$$

and assume static output feedback $u(t) = K\ y(t) = KC\ x(t)$. The above gain matrix has been selected so that it guarantees a certain control performance. Suppose now that due to system component failures (e.g. actuator or sensor failure/loss) or operating condition changes, the previous state-space representation can no longer model the dynamics of the new/impaired plant, which is now described by

$$\dot{x}(t) = A_f\, x(t) + B_f\, u(t), \qquad\qquad y(t) = C_f\, x(t) \tag{2}$$

Our objective is to design fast a new stabilizing output feedback control law $u(t) = K_f\ y(t) = K_f C_f\ x(t)$, so that the new closed-loop system $A_f + B_f K_f C_f$ captures as much of the dynamics of the nominal closed-loop system $A + BKC$ as possible. Hence, we need to find a new gain matrix that minimizes the Frobenius norm of the difference between the nominal and the new closed-loop system transition matrices. Therefore, the minimizing quantity is

$$\begin{aligned} J_{11} &= ||A + BKC - A_f - B_f K_f C_f||_F^2 \\ &= Tr[\, (A + BKC - A_f - B_f K_f C_f)^T (A + BKC - A_f - B_f K_f C_f)\,] \end{aligned} \tag{3}$$

where $||A||_F$ and $Tr(A)$ denote the Frobenius norm and the trace of a matrix A respectively. We know that given an asymptotically stable matrix A and a symmetric positive definite matrix Q, there exists a unique symmetric positive definite matrix P, such that $A^T P + PA + Q = 0$. The new gain matrix K_f needs to be stabilizing, that is it has to make $A_f + B_f K_f C_f$ stable. Therefore, according to above, it suffices to satisfy the following Lyapunov equation

$$\hat{A}_f^T P + P\hat{A}_f + Q = 0 \tag{4}$$

where $\hat{A}_f = A_f + B_f K_f C_f$. By including (4) in (3), we have a constrained minimization problem. Therefore, the minimizing quantity is given by $J_1 = J_{11} + Tr[\ L_1\ (\hat{A}_f^T P + P\hat{A}_f + Q)\ ]$, where $L_1 \in \Re^{n \times n}$ is the Lagrange multiplier matrix. In the analysis above, there is the underlying assumption that we know exactly the state-space matrices of the impaired model. This is not usually the case in applications, where we can only approximate the state-space matrices of the impaired system in cases of abrupt operating condition changes or severe failures. Hence, it is imperative that we design a controller that stabilizes the closed-loop system, even in the presence of uncertainty in some or all the state-space matrices $\{A_f, B_f, C_f\}$ of the closed-loop system $A_f + B_f K_f C_f$. We need the following theorem from [5]

Theorem 2.1 *Consider $\dot{x}(t) = Ax(t)$ with A a stability matrix; let $A^T P + PA + Q = 0$. Suppose that $A \to A + \Delta A$, then $\dot{y}(t) = (A + \Delta A)\, y(t)$ remains asymptotically stable if*

$$(\Delta A)\, Q^{-1}\, (\Delta A)^T \, < \, \frac{1}{4}\, P^{-1}\, Q\, P^{-1} \quad or \quad (\Delta A)^T\, P\, Q^{-1}\, P\, (\Delta A) \, < \, \frac{1}{4}\, Q \tag{5}$$

We can easily see that a sufficient condition for (5) to hold is

$$\sigma_{max}(\Delta A) < \frac{1}{2}\, \frac{\sigma_{min}(Q)}{\sigma_{max}(P)} \tag{6}$$

We can use (6) for $A = \hat{A}_f$ to maintain stability in cases of inevitable uncertainties in the closed-loop system $A_f + B_f K_f C_f$. Since Q is selected beforehand, in order to maximize the stability bound of (6), we simply need to minimize $\sigma_{max}(P)$. Since $\sigma^2_{max}(A) \leq ||A||^2_F = Tr(A^T A)$ for any matrix A, we choose to minimize $J_2 = Tr(P^T P) = Tr(P^2)$. Considering J_1 and J_2, the overall minimizing quantity is finally given by

$$J = Tr[\ (\tilde{A}_f - B_f K_f C_f)^T\ (\tilde{A}_f - B_f K_f C_f)\ +\ L_1\ (\hat{A}_f^T P + P\hat{A}_f + Q)\ +\ P^2\ ] \qquad (7)$$

where $\tilde{A}_f = A + BKC - A_f$. We compute the following gradients of the final cost J

$$\frac{\partial\ J}{\partial L_1} = \Delta_{L_1} \ =\ \hat{A}_f^T P + P\hat{A}_f + Q \qquad (8)$$

$$\frac{\partial\ J}{\partial P} = \Delta_P \ =\ \hat{A}_f L_1^T + L_1^T \hat{A}_f^T + 2P \qquad (9)$$

$$\frac{\partial\ J}{\partial K_f} = \Delta_{K_f} \ =\ 2B_f^T B_f K_f C_f C_f^T - 2B_f^T \tilde{A}_f C_f^T + B_f^T P(L_1 + L_1^T)C_f^T \qquad (10)$$

These gradients can be used by the algorithm of [11] to minimize (7). Note that this algorithm uses a version of the Broyden family method of conjugate directions, which is based on the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update rule; a version of this algoritm has been used in [12] for the design of controllers for robust stability and optimal performance of uncertain discrete-time systems.

Remark 2.1 The minimizing quantity of (7) consists of two components, the reconfiguration term (J_1) and the robustness term (J_2). By assigning weights to these terms, we could emphasize the one that is of more interest to us. Without loss of generality, we can always consider a weight $\omega_1 = 1$ for J_1, so that the minimizing quantity is given by $J_w = J_1 + \omega_2 J_2$. Note that the introduction of ω_2 affects only the gradient of (9), where the term $2\ P$ needs to be substituted by $2\ \omega_2 P$.

Remark 2.2 We have considered the robustness of the closed-loop system $\hat{A}_f$ assuming possible uncertainties in all the state-space matrices of the impaired system. If we are certain for some of these matrices, then the bound of (6) can be easily modified. For instance, let's consider a common case in reconfigurable systems, where the state and output matrices remain the same $A = A_f$, $C = C_f$ and we only have changes in B. Then, the allowable perturbations in B_f are given from (6) by $\sigma_{max}(\Delta B_f)\ <\ \frac{1}{2}\frac{\sigma_{min}(Q)}{\sigma_{max}(P)\ \sigma_{max}(K_f C_f)}$. By minimizing $\sigma_{max}(K_f C_f)$, we can further enlarge the above stability region. This can easily be done by including its upper bound $Tr[\ (K_f C_f)^T (K_f C_f)\ ]$ in the minimizing quantity J.

Remark 2.3 We may also wish to include in J another term that establishes a specific control performance, such as the familiar LQR cost $J_3' = \int_0^\infty (x^T Q_1 x + u^T R_1 u)\ dt$, where Q_1, R_1 are positive definite matrices of appropriate dimensions. Finally note that the results presented here can easily be extended to the dynamic output feedback case, [12].

3 An illustrative example

Consider an aircraft longitudinal control system from [10], with the linearized dynamic model

$$
\begin{aligned}
\begin{pmatrix} \dot{\alpha}(t) \\ \dot{\beta}(t) \\ \dot{\psi}(t) \\ \dot{\theta}(t) \end{pmatrix} &= \begin{pmatrix} -0.0582 & 0.0651 & 0 & -0.171 \\ -0.303 & -0.685 & 1.109 & 0 \\ -0.0715 & -0.658 & -0.947 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha(t) \\ \beta(t) \\ \psi(t) \\ \theta(t) \end{pmatrix} \\
&+ \begin{pmatrix} 0 & 1 \\ -0.0541 & 0 \\ -1.11 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \eta(t) \\ \tau(t) \end{pmatrix} \\
y(t) &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha(t) \\ \beta(t) \\ \psi(t) \\ \theta(t) \end{pmatrix}
\end{aligned} \tag{11}
$$

where $\alpha(t)$ and $\beta(t)$ are the forward and vertical speeds, $\psi(t)$ is the pitch rate and $\theta(t)$ is the pitch angle. The control inputs $\eta(t)$ and $\tau(t)$ are the elevator angle and throttle position respectively. When we consider static output feedback, the controller that assigns the closed-loop eigenvalues at $\{-2, -0.5973, -1.5 \pm 2j\}$ is given by

$$
K = \begin{pmatrix} -0.00031 & 4.77004 & 1.70457 \\ -2.01505 & -1.13002 & 0.02904 \end{pmatrix} \tag{12}
$$

Next, we suppose that the system dynamics change due to operating condition variations and at the same time severe failures happen at the actuators or sensors. First, we study the case of actuator loss and then the case of both actuator and sensor losses.

3.1 Actuator loss

The state-space matrices of the impaired model are given below. Note the loss of the second actuator.

$$
\begin{aligned}
A_f &= \begin{pmatrix} -0.0582 & 0.10 & 0.0 & -0.171 \\ -0.103 & -0.685 & 1.109 & 0 \\ -0.0715 & -0.658 & 1.98 & 0 \\ 0 & 0 & 1.5 & 0 \end{pmatrix}, \quad B_f = \begin{pmatrix} 0 & 0.0 \\ -0.09 & 0.0 \\ -1.11 & 0.0 \\ 0 & 0.0 \end{pmatrix} \\
C_f &= \begin{pmatrix} 0.9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.7 \\ 0 & 0 & 1 & 0 \end{pmatrix}
\end{aligned} \tag{13}
$$

Note that the state-space matrices given above correspond to new operating conditions, as given in [10]; in addition, we imposed the loss of one of the actuators. We use the algorithm of [12] for different weights to the robustness term, so that the minimizing quantity is given by J_w. In Table 1, we give the closed-loop eigenvalues and the first row of the stabilizing output feedback gain for the original and several impaired models. Note that for the impaired models the second row of the controller becomes irrelevant due to the actuator loss. In Table 2, we give the results for the reconfiguration and the robustness terms, and the robustness bound, that is the maximum singular value of the variations of the closed-loop system that can be allowed so that the asymptotic stability is maintained. We also restrict these variations in perturbations in the input matrix B

model	eigenvalues	gain K (first row)
nominal	$\{-2.0, -0.5973, -1.50 \pm 2.00j\}$	[-0.00031 4.77004 1.70458]
imp ($\omega_2 = 0.01$)	$\{-0.0770, -0.5591, -1.4610 \pm 2.5498j\}$	[0.00680 6.79671 4.31982]
imp ($\omega_2 = 0.1$)	$\{-0.0797, -0.5576, -1.4603 \pm 2.5491j\}$	[-0.06511 6.79624 4.31961]
imp ($\omega_2 = 1$)	$\{-0.0947, -0.5491, -1.4575 \pm 2.5352j\}$	[-0.45799 6.75011 4.32028]
imp ($\omega_2 = 10$)	$\{-0.1371, -0.5194, -1.4840 \pm 2.4060j\}$	[-1.37154 6.31477 4.37946]
imp ($\omega_2 = 50$)	$\{-0.1779, -0.4819, -1.6283 \pm 2.1596j\}$	[-1.92021 5.75013 4.64244]

Table 1: Actuator loss: eigenvalues and feedback gain for the nominal and the impaired models.

impaired model	J_1 (reconfiguration term)	J_2 (robustness term)	bound (general)	bound for ΔB_f
$\omega_2 = 0.01$	5.7469	2.3324	0.0668	0.0104
$\omega_2 = 0.1$	5.7534	2.2092	0.0687	0.0107
$\omega_2 = 1$	5.9737	1.7149	0.0785	0.0122
$\omega_2 = 10$	7.8258	1.1772	0.0956	0.0151
$\omega_2 = 50$	10.3066	1.0538	0.1012	0.0159

Table 2: Actuator loss: results for different weight factors considered for the impaired model.

and compute the same bound for $\sigma_{max}(\Delta B_f)$. Note that the same can easily be done for the case of structured perturbations in B_f. It is quite obvious that for larger ω_2, we enhance the robustness of the closed-loop system, which translates into smaller J_2 and larger perturbation bounds (for ΔB_f or general); at the same time, however, the reconfiguration term J_1 increases, which results in the deterioration of the closeness of the impaired closed-loop system $A_f + B_f K_f C_f$ to the nominal closed-loop system $A + BKC$.

In Fig. 1- 2, we compare the state-responses of the original and the impaired closed-loop systems. For the latter, we have chosen the controller obtained by our algorithm for $\omega_2 = 1$. The initial conditions vector was chosen as $(\,0.1 \quad 0.5 \quad 0.3 \quad 1\,)^T$. The two plots are very similar, which implies that despite the severity of the actutator loss, we were able to recover quite successfully the dynamics of the nominal model.

3.2 Actuator/sensor losses

We consider the even more severe case of losing both the second actuator and the first sensor. Hence, A_f and B_f remain the same as before, whereas the output matrix changes to

model	eigenvalues	gain K (first row)
nominal	$\{-2.0, -0.5973, -1.50 \pm 2.00j\}$	[-0.00031 4.77004 1.70458]
imp ($\omega_2 = 0.01$)	$\{-0.0773, -0.5589, -1.4610 \pm 2.5497j\}$	[-0.00031 6.79669 4.31982]
imp ($\omega_2 = 1$)	$\{-0.0773, -0.5590, -1.4594 \pm 2.5527j\}$	[-0.00031 6.80620 4.31707]
imp ($\omega_2 = 10$)	$\{-0.0772, -0.5598, -1.4488 \pm 2.5762j\}$	[-0.00031 6.88535 4.29861]

Table 3: Actuator/sensor losses: eigenvalues/feedback gain of original/impaired models.

impaired model	J_1 (reconfiguration term)	J_2 (robustness term)	bound (general)
$\omega_2 = 0.01$	5.7471	2.3202	0.0670
$\omega_2 = 1$	5.7472	2.3200	0.0670
$\omega_2 = 10$	5.7525	2.3190	0.0670

Table 4: Actuator/sensor losses: different weight factors for the impaired model.

$$\tilde{C}_f = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.7 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{14}$$

Note that these losses make the second row and the $(1,1)$ element of the first row of the controller gain irrelevant. In Tables 3-4, we give the same results as before for the present case. Comparing with the actuator loss case, we see that the sensor loss, in addition to the actuator loss, affects the robustness of the closed-loop system. Unlike the actuator case, we can not remove the pole at -0.0773, see Table 3, no matter how much we increase ω_2 in the minimizing quantity; in the actuator loss case we were able to enhance the robustness of the closed-loop system by assigning a large weight to the robustness term J_2, which resulted in removing the problematic pole from -0.0770 to -0.1779. Comparing the reconfiguration terms J_1 of Tables 2 and 4, we see that the loss of the first sensor did not affect at all the reconfiguration aspect of our design. Finally note that the state-responses of the impaired closed-loop system after the loss of both the actuator and the sensor for the same initial vector as before are almost identical to the ones of Fig. 2, which shows that despite the loss of the sensor, in addition to the actuator loss, our scheme was capable of recovering the dynamics of the original system.

In Table 5, we compare the Frobenius norm of the difference between the original and the impaired closed-loop transition matrices $||A + BKC - A_f - B_f K_f C_f||_F$ for the controllers derived in the examples of [7] and [10] and the ones derived by the proposed algorithm here for the cases of $\omega_2 = 0.1$ and $\omega_2 = 1$. The present approach, in addition to maintaining closed-loop stability, is more successful in preserving the characteristics of the original system compared to the techniques presented in the papers above.

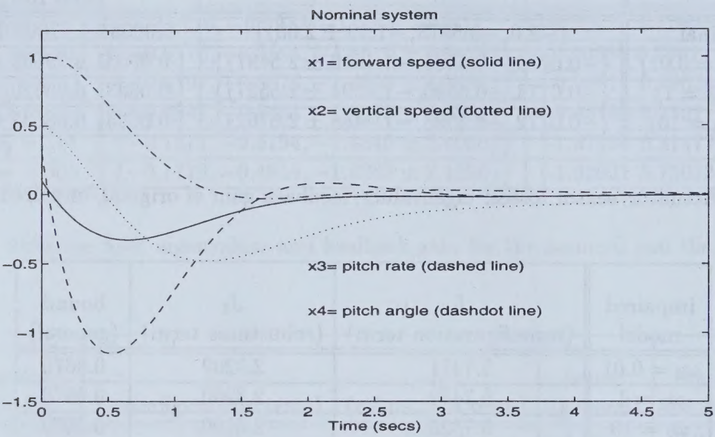


Figure 1: Nominal system: state trajectories for the closed-loop system.

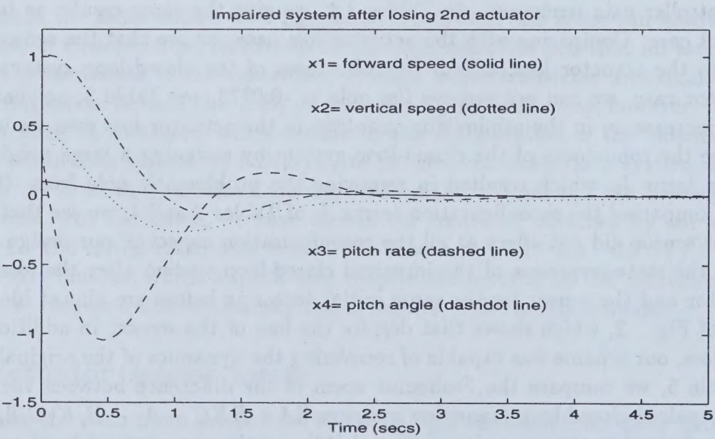


Figure 2: Actuator loss: state trajectories for the impaired closed-loop system.

example of Gao *et al* (1991)	present algor. $\omega_2 = 1$	present algor. $\omega_2 = 0.1$	example of Jiang (1994)	present algor. $\omega_2 = 1$	present algor. $\omega_2 = 0.1$
3.8525	2.7038	2.4225	2.4627	0.6392	0.6392

Table 5: Comparing $||A + BKC - A_f - B_f K_f C_f||_F$ for literature examples.

4 Conclusions

The problem of control reconfiguration in response to operating condition changes or abrupt system component failures has been studied here. An optimization algorithm has been presented that provides an output feedback controller that not only stabilizes the new/impaired system but also preserves much of the dynamics of the original/unfailed system. The design is such that the closed-loop system is robust with respect to uncertainties/modelling errors in the state-space model of the impaired system.

Although the interest here is in continuous-time systems, a similar approach can be applied to the discrete-time case. Results concerning eigenstructure assignment can be included in our design so that additional restrictions/specifications can be introduced in the minimizing quantity with respect to the eigenspace where the closed-loop eigenvectors are desired/needed to vary. In that respect, the requirement of closeness to the original closed-loop system can be specialized to the case where the closeness to the eigenstructure of the original system becomes the main objective of the reconfiguration design.

References

[1] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Applications.* New York: John Wiley & Sons, Inc., 1993.

[2] M. Bodson, " Identification with modeling uncertainty and reconfigurable control, " in *Proc. 32nd Conf. Decision Contr.*, San Antonio, TX, pp. 2242-2247, 1993.

[3] A. K. Caglayan, S. M. Allen, and K. Wehmuller, " Evaluation of a second generation reconfiguration strategy for aircraft flight control systems subjected to actuator failure/surface damage, " in *Proc. IEEE 1988 Nat. Aerosp. Electron. Conf.*, pp. 520-529, 1988.

[4] P. Chandler, M. Mears, and M. Pachter, " On-line optimizing networks for reconfigurable control, " in *Proc. 32nd Conf. Decision Contr.*, San Antonio, TX, pp. 2272-2277, 1993.

[5] M. Eslami and D. L. Russell, " On stability with large parameter variations: stemming from the direct method of Lyapunov, " *IEEE Trans. Automat. Contr.*, vol. 25, no. 6, pp. 1231-1234, 1980.

[6] R. A. Eslinger and P. R. Chandler, " Self-repairing control system program overview, " in *Proc. IEEE 1988 Nat. Aerosp. Electron. Conf.*, pp. 504-511, 1988.

[7] Z. Gao and P. J. Antsaklis, " Stability of the pseudo-inverse method for reconfigurable control systems, " *Int. J. Contr.*, vol. 53, no. 3, pp. 717-729, 1991.

[8] V. F. Gavito and D. J. Collins, " Application of eigenstructure assignment to self reconfiguring aircraft MIMO controllers, " in *Proc. AIAA Guidance, Navig., Contr. Conf.*, pp. 1-12, 1987.

[9] P. Howell, " Full authority lateral-directional flight control system design using a reconfigurable control and adaptive learning network: a design proposal, " in *Proc. 1st IEEE Conf. Contr. Appl.*, Dayton, OH, pp. 860-865, 1992.

[10] J. Jiang, " Design of reconfigurable control systems using eigenstructure assignments, " *Int. J. Contr.*, vol. 59, no. 2, pp. 395-410, 1994.

[11] I. K. Konstantopoulos and P. J. Antsaklis, " An optimization strategy for reconfigurable control systems, " *Technical Report of the ISIS Group at the University of Notre Dame, ISIS-95-006*, September 1995.

[12] I. K. Konstantopoulos and P. J. Antsaklis, " Optimal design of robust controllers for uncertain discrete-time systems, " *Int. J. Contr.*, to appear.

[13] D. P. Looze, J. L. Weiss, J. S. Eterno, and N. M. Barrett, " An automatic redesign approach for restructurable control systems, " *IEEE Control Syst. Mag.*, vol. 5, no. 2, pp. 16-22, 1985.

[14] D. D. Moerder, N. Halyo. J. R. Broussard, and A. K. Caglayan, " Application of precomputed control laws in a reconfigurable aircraft flight control system, " *J. Guidance, Contr., Dynamics*, vol. 12, no. 3, pp. 325-333, 1989.

[15] A. J. Ostroff and R. M. Hueschen, " Investigation of control law reconfigurations to accommodate a control element failure on a commercial airplane, " in *Proc. 1984 Amer. Contr. Conf.*, San Diego, CA, pp. 1746-1754, 1984.

[16] A. J. Ostroff, " Techniques for accommodating control effector failures on a mildly statically unstable airplane, " in *Proc. 1985 Amer. Contr. Conf.*, Boston, MA, pp. 906-913, 1985.

[17] K. S. Rattan, "Evaluation of control mixer concept for reconfiguration of flight control system, " in *Proc. IEEE 1985 Nat. Aerosp. Electron. Conf.*, pp. 560-569, 1985.

[18] K. M. Sobel, E. Y. Shapiro, and A. N. Andry, " Eigenstructure assignment, " *Int. J. Contr.*, vol. 59, no. 1, pp. 13-37, 1994.

[19] Y. Y. Wang and N. E. Wu, " An approach to reconfiguration of robust control systems for robust failure detection, " in *Proc. 32nd Conf. Decision Contr.*, San Antonio, TX, pp. 1704-1709, 1993.

[20] N. E. Wu, " Reconfigurable control design: achieving stability robustness and failure tracking, " in *Proc. 32nd Conf. Decision Contr.*, San Antonio, TX, pp. 2278-2283, 1993.

On the H_∞ Control of Bilinear Systems Using Time Domain Approach

Shyh-Pyng Shue, Graduate Fellow, Department of Aerospace Engineering
M. Edwin Sawan, Professor, Department of Electrical Engineering
Kamran Rokhsaz, Assistant Professor, Department of Aerospace Engineering
Wichita State University
Wichita, KS 67260

ABSTRACT

Applying H_∞ control by time domain approach to bilinear systems is presented in this paper. Two Hamilton-Jocobi-Bellman equations are employed to find the optimality formulae for H_∞ output feedback optimal or suboptimal control of bilinear systems. It is shown that the difference between linear systems and bilinear systems by using the H_∞ control law is in the fact of that bilinear systems need one more condition for both the state feedback case and the dynamic output feedback case. As this condition is also satisfied, like linear systems, two quadratic Lyapunov functions can be found so that the nonlinear closed loop systems are globally asymptotically stable by using the H_∞ control law. A single input bilinear system representing the dryer section of a paper-making machine in a paper mill is employed to demonstrate the method.

Keywords: Bilinear Systems, H_∞ Robust Control, HJB Inequalities, State Feedback Control, Output Feedback Control, and Quadratic Lyapunov Functions.

I. INTRODUCTION

H_∞ optimal control theory has attracted a lot of attention during the past decade. The early important contributions were made by Zames [1], and Francis and Doyle [2] [3]. The standard state feedback H_∞ method was formulated by Petersen [4]. The H_∞ optimal control theory by output feedback was presented by Doyle et. al. [5]. However, all of the above contributions are limited to linear systems, because these methods rely on norm-based approach in the frequency domain to find the minimum value of γ.

However, the nonlinear and bilinear H_∞ optimal control methods can not be formulated in the frequency domain. Therefore, the time domain approach for nonlinear and bilinear H_∞ optimal control is necessary. Examples of such formulation for nonlinear systems can be found in Ref. [6-9], but none is given for bilinear systems. In this paper, H_∞ optimal control of bilinear systems is presented for both state feedback and output feedback cases. Like nonlinear cases, the bilinear H_∞ optimal control problem by using output feedback is related to two Hamilton-Jacobi-Bellman (HJB) equations, known as Isaacs equations. These two HJB equations, by properly selecting the quadratic Lyapunov function, can be simplified to two algebraic Riccati equations. The conditions for H_∞ optimal control of bilinear systems to have globally asymptotically stable closed loop systems are also specified for the state feedback case and the output feedback case.

II. METHOD OF ANALYSIS

1. PROBLEM STATEMENT AND ASSUMPTIONS

From Ref. [10] and [11], multi-input bilinear systems can be decomposed into a number of single input subsystems with individual inputs. Therefore, if one of these single input bilinear systems can be controlled, then the multi-input bilinear system can also be controlled. Therefore, single input bilinear systems are discussed here instead of multi-input bilinear systems.

Consider a single input time invariant bilinear system as follows:

$$\begin{aligned}&\dot{x} = Ax + (B_0 + B_1 x)u + Gw_1, \quad x(0) = x_0\\&z = C_1 x + D_{12} u \qquad (1)\\&y = C_2 x + D_{21} w_2,\end{aligned}$$

where x represents the system state variables, y is the system measured output, z is the system control output, u is the system control, $w(t) = \left[w_1 \quad w_2\right]^T$ are exogenous inputs, and A, B_0, B_1, G, C_1 and C_2 have proper dimensions, respectively. In order to simplify the H_∞ bilinear problem and provide a reasonable solution, the following assumptions need to be made [5],

1) $D_{12}^T \cdot C_1 = 0$
2) $D_{12}^T \cdot D_{12} = I$
3) $G \cdot D_{21}^T = 0$
4) $D_{21} \cdot D_{21}^T = I$.

The first and second conditions indicate that there is no cross term between the penalty variable, z, and the control input, u, and the control weight matrix is the identity matrix. Likewise, the third and fourth conditions describe the similar relations between the measured output, y, and exogenous inputs, w.

2. METHODOLOGY FOR STATE FEEDBACK

The state feedback H_∞ suboptimal control method involves finding a state feedback control,

$$u = u(x), \text{ and } w_1 = w_1(x) \tag{2}$$

such that the following equation holds,

$$\int_0^T \|z(s)\|^2 \, ds \leq \gamma^2 \int_0^T \|w(s)\|^2 \, ds, \tag{3}$$

where $0 < t < T$. Using (3) as the penalty function in combination with the closed loop Lyapunov function of the system, $V(x)$, the Hamilton-Jacobi-Bellman (HJB) equation can be obtained as follows:

$$H = V(x(T)) - V(x(0)) + \frac{1}{2}\int_0^T \|z(s)\|^2 ds - \frac{\gamma^2}{2}\int_0^T \|w(s)\|^2 ds \leq 0. \tag{4}$$

Computing the derivative with respect to time and using the above first and second simplifying assumptions, results in the following HJB equation,

$$\begin{aligned} H = {} & V_x(x)Ax + V_x(x)(B_0 + B_1 x)u + V_x(x)Gw_1 \\ & + \frac{1}{2}x^T C_1^T C_1 x + \frac{1}{2}u^T u - \frac{\gamma^2}{2} w_1{}^T w_1 \leq 0, \end{aligned} \tag{5}$$

where $V_x(x) = \partial V(x)/\partial x$. Setting derivative with respect to u equal to zero, results in the suboptimal control which is

$$u = -(B_0 + B_1 x)^T V_x^T. \tag{6}$$

Similarly, setting

$$\frac{\partial H}{\partial w_1} = 0, \tag{7}$$

results in

$$w_1 = \frac{1}{\gamma^2} G^T V_x^T. \tag{8}$$

Substituting (6) and (8) back into (5), results in

$$\begin{aligned} H = {} & V_x(x)Ax + \frac{1}{2}x^T C_1^T C_1 x - \frac{1}{2}V_x(x)(B_0 + B_1 x)(B_0 + B_1 x)^T V_x^T \\ & + \frac{1}{2\gamma^2} V_x(x) G G^T V_x^T(x) \leq 0. \end{aligned} \tag{9}$$

In Ref. [10] and [11], the closed loop Lyapunov function for bilinear systems is shown to be a quadratic function of the state vector. Similarly, suppose that

$$V(x) = \frac{1}{2}x^T X x. \tag{10}$$

Therefore, the HJB equation for H_∞ control of bilinear systems for the state feedback case is

$$H = x^T XAx + \frac{1}{2}x^T C_1^T C_1 x - \frac{1}{2}x^T X(B_0 + B_1 x)(B_0 + B_1 x)^T Xx$$
$$+ \frac{1}{2\gamma^2}x^T XGG^T Xx \le 0 \tag{11}$$

Like Ref. [11], the solution of equation (11), X, can be shown to satisfy a Hamiltonian matrix,

$$H_\infty = \begin{bmatrix} A & -B_0 B_0^T + \frac{1}{\gamma^2}GG^T \\ -C_1^T C_1 & -A^T \end{bmatrix}. \tag{12}$$

However, in order to guarantee equation (11) is satisfied by the Hamiltonian matrix (12), the following condition must be satisfied [11],

$$C_1^T C_1 - XB_0 B_0^T X \ge 0 \text{ if } B_1 x B_0^T \ne 0. \tag{13}$$

The Hamiltonian matrix (12) is associated with the following algebraic Riccati equation,

$$A^T X + XA + C_1^T C_1 - X(B_0 B_0^T - \frac{1}{\gamma^2}GG^T)X = 0. \tag{14}$$

Therefore, the following theorem can be established.

Theorem 2.1:

For single input bilinear systems, the closed loop nonlinear systems from H_∞ state feedback control are globally asymptotically stable if and only if the following conditions are satisfied,

1) (A, C_1) is detectable
2) (A, B_0) is stabilizable
3) (A, G) is stabilizable
4) and $C_1^T C_1 - XB_0 B_0^T X \ge 0$, if $B_1 x B_0^T \ne 0$.

Proof of theorem 2.1 is given in Ref. [5] and [11].

Applying this method, the associated closed loop system using H_∞ state feedback control becomes

$$\dot{x} = Ax - (B_0 + B_1 x)(B_0 + B_1 x)Xx + \frac{1}{\gamma^2}GG^T Xx. \tag{15}$$

There is an example in the next section which will explain the method very clearly. In fact, if the fourth condition is violated, the closed loop nonlinear system can still be stable in the small.

3. METHODOLOGY FOR DYNAMIC OUTPUT FEEDBACK

In order to provide an estimate for the state variables, the following observer designed based on the system in equation (1) with its measured output, y, is needed,

$$\dot{\hat{x}} = A\hat{x} + (B_0 + B_1\hat{x})u + K(y - C_2\hat{x}), \ \hat{x}(0) = x_0, \tag{16}$$

where K is a suitable matrix. The constant value of K will be shown later. The design of output feedback is based on these estimated state variables,

$$u = u(\hat{x}), \ w_1 = w_1(\hat{x}), \text{ and } w_2 = w_2(\hat{x}), \tag{17}$$

such that the following HJB equation based on equations (3) and (16) is satisfied,

$$H = V_x\dot{x} + V_{\hat{x}}\dot{\hat{x}} + \frac{1}{2}x^T C_1^T C_1 x + \frac{1}{2}\|u(\hat{x})\|^2 - \frac{\gamma^2}{2}\|w_1(\hat{x})\|^2 - \frac{\gamma^2}{2}\|w_2(\hat{x})\|^2 \le 0. \tag{18}$$

Following the same procedure as in the previous case,

$$w_2 = \frac{1}{\gamma^2}K^T V_{\hat{x}}^T. \tag{19}$$

Substituting system in equation (1) and equations (16) and (19) back into equation (18), the HJB equation becomes

$$H = V_x\big(Ax + (B_0 + B_1 x)u(\hat{x})\big) + V_{\hat{x}}\big(A\hat{x} + (B_0 + B_1\hat{x})u(\hat{x}) + K(C_2 x - C_2\hat{x})\big)$$
$$+\frac{1}{2}x^T C_1^T C_1 x + \frac{1}{2}u(\hat{x})^T u(\hat{x}) + \frac{1}{2\gamma^2}V_x G G^T V_x^T + \frac{1}{2\gamma^2}V_{\hat{x}}K^T K V_{\hat{x}}^T \le 0 \tag{20}$$

Because the closed loop Lyapunov functions of bilinear systems have quadratic forms from [10] [11], one can therefore assume that

$$V(x,\hat{x}) = \frac{1}{2}x^T X x + \frac{\gamma^2}{2}(x-\hat{x})^T Z^{-1}(x-\hat{x}) \tag{21}$$

where

$$Z = Y(1 - \frac{1}{\gamma^2}XY)^{-1}, \tag{22}$$

and X, Y, and Z are symmetric positive definite matrices, that means $XY \le \gamma^2 I$. Therefore,

$$V_x = x^T X + \gamma^2(x-\hat{x})^T Z^{-1}, \tag{23}$$

and

$$V_{\hat{x}} = -\gamma^2(x-\hat{x})^T Z^{-1}. \tag{24}$$

From the state feedback case, one can assume that

$$u(\hat{x}) = -(B_0 + B_1\hat{x})^T X\hat{x}. \tag{25}$$

Equation (20) must be true, no matter what x and $\hat{x}$ are. In order to simplify equation (20), substituting equations (6), (23), (24), and (25) back into (20) and setting $x = \hat{x}$ implies

$$V_x = x^T X, \quad V_{\hat{x}} = 0 \text{ and } u(\hat{x}) = u(x), \tag{26}$$

resulting in

$$\begin{aligned} H = {} & x^T XAx + \frac{1}{2}x^T C_1^T C_1 x + \frac{1}{2}x^T XGG^T Xx \\ & - \frac{1}{2}x^T X(B_0 + B_1 x)(B_0 + B_1 x)^T Xx \le 0. \end{aligned} \tag{27}$$

Because the last term of the above equation is positive definite, like the previous case, the solution, X, of equation (27) satisfies the Hamiltonian matrix,

$$H_\infty = \begin{bmatrix} A & -B_0 B_0^T + \dfrac{1}{\gamma^2}GG^T \\ -C_1^T C_1 & -A^T \end{bmatrix}. \tag{28}$$

In fact, the above equation is the same as the one for the state feedback case. The conditions for the closed loop system being globally asymptotically stable should also be the same. Similarly, setting $\hat{x} = 0$, results in

$$V_x = \gamma^2 x^T Y^{-1} \text{ and } V_{\hat{x}} = -\gamma^2 x^T Z^{-1}. \tag{29}$$

From equation (25), because $\hat{x} = 0$, $u(\hat{x}) = 0$. The HJB equation therefore becomes

$$\begin{aligned} H = {} & \gamma^2 x^T Y^{-1} Ax + \frac{1}{2}x^T C_1^T C_1 x + \frac{\gamma^2}{2}x^T Y^{-1} GG^T Y^{-1} x - \frac{\gamma^2}{2}x^T C_2^T C_2 x \\ & + \frac{1}{2\gamma^2}\left\| \gamma^2 C_2 x + K^T V_{\hat{x}}^T \right\|^2 \le 0 \end{aligned} \tag{30}$$

Now, assuming

$$K^T V_{\hat{x}}^T = -\gamma^2 C_2 x, \tag{31}$$

results in

$$K = Z C_2^T. \tag{32}$$

The solution of equation (30), Y, satisfies the following Hamiltonian matrix,

$$J_\infty = \begin{bmatrix} A^T & \dfrac{1}{\gamma^2}C_1^T C_1 - C_2^T C_2 \\ -GG^T & -A \end{bmatrix}. \tag{33}$$

This means that Y satisfies the following algebraic Riccati equation,

$$AY + YA^T + GG^T - Y(C_2^T C_2 - \frac{1}{\gamma^2}C_1^T C_1)Y = 0. \tag{34}$$

Theorem 2.2:

For single input bilinear systems, the closed loop systems resulting from dynamic output feedback control are asymptotically stable in the large if and only if the following conditions hold,

1) $(A,\ C_1)$ and $(A,\ C_2)$ is detectable,
2) $(A,\ B_0)$ is stabilizable,
3) $(A,\ G)$ is stabilizable
4) $C_1^T C_1 - XB_0B_0^T X \geq 0$, if $B_1xB_0^T \neq 0$,
5) and $XY \leq \gamma^2 I$.

Now, the closed loop system of this observer becomes

$$\dot{\hat{x}} = A\hat{x} - (B_0 + B_1\hat{x})(B_0 + B_1\hat{x})^T X\hat{x} + Y(1 - \frac{1}{\gamma^2}XY)^{-1} C_2^T (y - C_2\hat{x}), \quad (35)$$

and the closed loop system is

$$\dot{x} = Ax - (B_0 + B_1x)(B_0 + B_1\hat{x})X\hat{x} + \frac{1}{\gamma^2}GG^T X\hat{x}. \quad (36)$$

Notice that the inequality of

$$GG^T - YC_2^T C_2 Y \geq 0 \quad (37)$$

does not need to be considered, because the corresponding equation in the open loop system does not have a bilinear term. In fact, the difference in H_∞ control between linear systems and bilinear systems is in condition (4). The closed loop nonlinear system will be globally asymptotically stable, if and only if condition (4) is also satisfied. Otherwise the closed loop system is asymptotically stable only in the small.

III. RESULTS AND DISCUSSION

1. ILLUSTRATION

The example for the H_∞ state feedback case is selected from Ref. [10]. The dryer section of a paper-making machine in a paper mill is described by the following second order single input bilinear system,

$$\dot{x} = Ax + (B_0 + B_1x)u \text{ and } Z = C_1x + u$$

where

$$A = \begin{bmatrix} -0.046 & 0 \\ -0.7632 & -3.197 \end{bmatrix}, \quad B_0 = \begin{bmatrix} 0.986 \\ 0 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} -0.27 & 0 \\ 0 & 0 \end{bmatrix}, \text{ and } C_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

where $x = [x_1 \;,\; x_2]^T$ represents state variables. Note that the penalty output equation is assumed by the authors. Reference [12] shows that one can assume some noise in x_2 only, that is

$$x_2 = \bar{x}_2 + w_1,$$

where $\bar{x}_2$ represents the value without noise. The bilinear system with noise can therefore be obtained as follows,

$$\dot{x} = Ax + (B_0 + B_1 x)u + Gw_1 \text{ and } Z = C_1 x + u,$$

where A, B_0, B_1, and C_1 are unchanged, and

$$G = \begin{bmatrix} 0 \\ -3.197 \end{bmatrix}.$$

Based on the above stochastic bilinear system, γ and the solution of the Hamiltonian matrix, X, can be found by using the bisection method as

$$\gamma = 0.9727, \text{ where } X = \begin{bmatrix} 2.1374 & -0.5140 \\ -0.5140 & 0.4331 \end{bmatrix}.$$

However, the last condition of theorem 2.1 is not satisfied when the above values are used. Enforcing the last condition of theorem 2.1, γ and X will be altered to at least the following values

$$\gamma = 1.2650, \text{ where } X = \begin{bmatrix} 1.0130 & -0.0493 \\ -0.0.0493 & 0.1934 \end{bmatrix}.$$

One can verify that the closed loop nonlinear system by using the H_∞ state feedback control law based on the above two values is globally asymptotically stable. Any value of γ larger than this will also satisfy the last constraint as well. However, larger values of γ result in a system less sensitive to noise. Note that the minimum value of γ for linear systems is not suitable if the global solution is required.

2. DISCUSSION

The problem of finding H_∞ optimal control for bilinear systems by state feedback is related to an algebraic Riccati equation. The conditions for the closed loop systems to have globally asymptotically stable solutions are the same as those of linear systems plus one more. The additional restriction is $C_1^T C_1 - XB_0 B_0^T X \geq 0$, if $B_1 x B_0^T \neq 0$. The time domain approach for the bilinear H_∞ optimal control problems is to find an HJB equation. This equation can be assumed to have a quadratic closed loop Lyapunov function such that the HJB equation can be simplified as an algebraic Riccati equation. Unlike the nonlinear case [6] [7] where the closed loop Lyapunov function is also quadratic, the present closed loop nonlinear system is asymptotically stable in the large.

The restrictions for H_∞ optimal output feedback control of bilinear systems to be globally asymptotically stable are similar to the previous situation. Although two algebraic Riccati equations are created, the effect of the second Riccati equation is identical to the linear case. However, only the Riccati equation created by state feedback needs to satisfy the additional constraint that $C_1^T C_1 - XB_0 B_0^T X \geq 0$ such that the HJB equation is satisfied.

IV. CONCLUSION

Optimal H_∞ control of bilinear systems by using the time domain approach was presented. It was shown that instead of two HJB equations two algebraic Riccati equations are needed. In order to guarantee global stability of the closed loop system, it was shown that one more condition in addition to those of the linear H_∞ optimal control problem needs to be satisfied. The equations for both state feedback and output feedback cases were derived. The bilinear system of the dryer section of a paper-making machine in a paper mill was used to illustrate the state feedback case.

REFERENCE

1. Zames, G. " Feedback and Optimal Sensitivity: Model Reference Transformations, Multiplicative Seminorms, and Approximate Inverse," *IEEE Transactions on Automatic Control*, Vol. 26, No. 2, pp. 301 - 320, 1981.
2. Francis, B. A., *A Course in H_∞ Controlled Theory*, New York, Springer-Verlag, 1986.
3. Francis, B. A., and Doyle, J. C., "Linear Control Theory with an H_∞ Optimality Criterion," *SIAM Journal on Control and Optimization,* Vol. 25, No. 4, pp. 815 - 844, 1987.
4. Pertersen, I. R., "Disturbance Attenuation and H_∞ Optimization: A Design Method Based on the Algebraic Riccati Equation," *IEEE Transactions on Automatic Control*, Vol. 32, No. 5, pp. 427 -429, May 1987.
5. Doyle, J. C., Glover, K., Khargonekar P. P., and Francis, B. A, "State-Space Solution to Standard H_2 and H_∞ Control Problems," *IEEE Transactions on Automatic Control*, Vol. 34, No. 8, pp. 831 -847, Aug. 1989.

6. Isidori, A., and Astolfi, A., "Disturbance Attenuation and H_∞-Control Via Measurement Feedback in Nonlinear Systems," *IEEE Transactions on Automatic Control*, Vol. 37, No. 9, pp. 1283 -1293, Sep. 1992.
7. van der Schaft, "Nonlinear H_∞ Control and Hamilton-Jacobi Inequalities," *IFAC Nonlinear Control Systems Design,* pp. 9-14, Bordeaux, France, 1992.
8. Isidori, A., " H_∞ Control Via Measurement Feedback for Affine Nonlinear Systems," *International Journal of Robust and Nonlinear Control,* Vol. 4, pp. 552-574, 1994.
9. Lu, Wei-Min and Doyle, J. C., "H_∞ Control of Nonlinear Systems Via Output Feedback: Controller Parameterization," *IEEE Transactions on Automatic Control*, Vol. 39, No. 12, pp. 2517 -2521, Dec. 1994.
10. Ying, Y. Q., Rao, M., and Sun, Y. X., "Suboptimal Control for Bilinear Systems," *Optimal Control Applications & Methods,* Vol. 14, pp. 195-202, 1993.
11. Jacobson, D. H., *Extensions of Linear-Quadratic Control, Optimization and Matrix Theory*, Mathematics in Science and Engineering, Vol. 133, New York, Academic Press, 1977, pp. 123-149.
12. McLEAN, Donald, *Automatic Flight Control Systems*, New York, Prentice Hall, 1990, pp. 138.

Robust Polyhedral Stability Via Extreme Points Results

F. Amato, A. Caruso, F. Garofalo and L. Glielmo

Dipartimento di Informatica e Sistemistica
Università degli Studi di Napoli "Federico II"
via Claudio 21, 80125 Napoli, Italy
Email: garofalo@disna.dis.unina.it

Abstract

In this paper we deal with linear, continuous-time systems depending on uncertain and time-varying parameters. When the system matrix depend on parameters as the ratio of multiaffine polynomials and the parameter bounding set is a hyperrectangle, we provide a procedure which allows to prove the exponential stability of the system under consideration using the class of *polyhedral* Lyapunov functions. Our main results are based on extreme points properties. Examples of applications are presented.

1 Introduction

In this paper we deal with the stability of the uncertain system

$$\dot{x}(t) = A(\pi(t))x(t)\,, \tag{1}$$

where the vector of uncertain parameters is any Lebesgue measurable function $\pi(\,\cdot\,) : [0, +\infty) \rightarrow \mathcal{R} \subset \mathbb{R}^{n_p}$. We assume that the parameter bounding set $\mathcal{R}$ is a hyperrectangle, i.e.

$$\mathcal{R} := [\underline{\pi}_1, \overline{\pi}_1] \times \cdots \times [\underline{\pi}_{n_p}, \overline{\pi}_{n_p}], \tag{2}$$

and that $A(\,\cdot\,)$ is the ratio of multiaffine polynomials in π

$$A(\pi) = \frac{\sum_{i_1,i_2,\ldots,i_{n_p}=0}^{1} A_{i_1 i_2 \ldots i_{n_p}} \pi_1^{i_1} \pi_2^{i_2} \cdots \pi_{n_p}^{i_{n_p}}}{\sum_{i_1,i_2,\ldots,i_{n_p}=0}^{1} a_{i_1 i_2 \ldots i_{n_p}} \pi_1^{i_1} \pi_2^{i_2} \cdots \pi_{n_p}^{i_{n_p}}} \tag{3}$$

where $A_{i_1 i_2 \ldots i_{n_p}} \in \mathbb{R}^{n \times n}$ and $a_{i_1 i_2 \ldots i_{n_p}} \in \mathbb{R}$. We also suppose that the denominator in (3) is always different from zero in $\mathcal{R}$.

Assume that system (1) is exponentially stable for any realization of the uncertainty $\pi(\,\cdot\,)$ ranging in $\mathcal{R}$ and that it admits common Lyapunov functions. Then, according to [4], we say that a certain class $\mathcal{C}$ of candidate Lyapunov functions is universal for the system if the system admits a Lyapunov function in $\mathcal{C}$. Following this definition, the "classical" time-invariant quadratic Lyapunov functions (see [8] and the bibliography therein) are not universal for system (1) as an example in [10] has shown, while *polyhedral* Lyapunov functions are (see [3] and [7]). This observation justifies the effort of searching for a polyhedral Lyapunov function for a given uncertain system. Typically (see for example [3]) the proposed procedures consider a discrete-time system and afterwards, via Eulero approximation, these are extended to handle continuous-time systems.

Here, instead, we proceed to the *direct* construction of a polyhedral Lyapunov function for a given continuous-time system. To this aim we will briefly introduce some properties of polyhedral sets and polyhedral functions and show that, for the problem of checking the negative definiteness of the derivative of a given polyhedral function, "extreme point" results hold (see [2]). An example illustrates our technique.

2 Preliminaries

In the sequel we shall use the following definition.

Definition 1 A polyhedron is a polytope with the origin in its interior, i.e.

$$\mathcal{P} = \operatorname{conv}\{x^{(1)}, \dots, x^{(m)}\}, \qquad 0 \in \operatorname{int}\{\mathcal{P}\} \tag{4}$$

where $x^{(1)}, \dots, x^{(m)} \in \mathbb{R}^n$ are the vertices of $\mathcal{P}$ and $\operatorname{int}\{\mathcal{P}\}$ denotes the interior of $\mathcal{P}$. In [11, p. 6], it is shown that a given polyhedron $\mathcal{P}$ can also be described by the intersection of a finite collection of halfspaces, i.e.

$$\mathcal{P} = \left\{ x : p_i^T x \leq 1 \,,\, i = 1, \dots, s \right\}, \tag{5}$$

where $p_i^T \in \mathbb{R}^n$, $i = 1, \dots, s$, are appropriate row-vectors.

We remind [11, p. 162] that the exposed faces of a convex set $\mathcal{S}$ are the sets of the form $\mathcal{S} \cap \mathcal{H}$, where $\mathcal{H}$ is a supporting hyperplane to $\mathcal{S}$. It is straightforward to show that every p_i^T describes a supporting hyperplane for $\mathcal{P}$ and that the exposed faces of the polyhedron are polytopes which are the convex hull of the vertices of $\mathcal{P}$ belonging to the same supporting hyperplane. In the following we talk of the i-th face of $\mathcal{P}$ referring to the exposed face associated to the i-th hyperplane $p_i^T x = 1$ and we denote by $P = (p_1 \dots p_s)^T$ the matrix of the row vectors p_i^T.

A given polyhedron $\mathcal{P}$ *induces* a "polyhedral function" as follows

$$v_{\mathcal{P}}(x) := \max_{i=1,\dots,s} p_i^T x \,. \tag{6}$$

From the definition it follows that a polyhedral function is positive definite, convex and enjoys a radial property, that is

$$v_{\mathcal{P}}(\mu x) = \mu v_{\mathcal{P}}(x) \,, \qquad \forall \mu \geq 0 \,. \tag{7}$$

As a consequence, it is straightforward to show that the level surfaces of $v_{\mathcal{P}}$ are the boundaries of polyhedra similar to $\mathcal{P}$ and viceversa, and that the polyhedron $\mathcal{P}$ is the level surface $v_{\mathcal{P}}(x) = 1$.

For more details on polyhedral functions the interested reader is referred to [3].

3 Stability Analysis of Certain Systems

In this section we deal with the linear certain system of the form

$$\dot{x} = Ax\,. \tag{8}$$

For the stability analysis of system (8) we shall utilize the following version of the Lyapunov Stability Theorem, whose proof can be found in [9, Ch. 5].

Fact 1 *Assume there exists a function* $v : \mathbb{R}^n \to \mathbb{R}$ *such that*

i) $v(\cdot)$ *is continuous;*

ii) $v(\cdot)$ *is positive definite;*

iii) the Dini (upper right) derivative of v *along the solutions of system (8), defined as*

$$Dv(x) = \limsup_{\tau \to 0^+} \frac{v(x + \tau\dot{x}) - v(x)}{\tau}\Big|_{\dot{x}=Ax} \tag{9}$$

is negative definite.

Then system (8) is exponentially stable.

A function v satisfying i) and ii) is said to be a candidate Lyapunov function. We will call (9) the Lyapunov derivative of v.

In this paper we shall consider candidate Lyapunov functions $v_{\mathcal{P}}(x)$ of polyhedral type; this makes sense because $v_{\mathcal{P}}$ is continuous and positive definite. According to the above discussion, we shall refer to $v_{\mathcal{P}}$ as a (candidate) *polyhedral Lyapunov function.*

The Lyapunov derivative of a polyhedral function $v_{\mathcal{P}}$ is given by (see [4])

$$Dv_{\mathcal{P}}(x) = \max_{j \in I(x)} p_j^T Ax\,, \tag{10}$$

where $I(x)$ is the set of the indeces j such that

$$v_{\mathcal{P}}(x) = p_j^T x\,. \tag{11}$$

It is straightforward to show that also the Lyapunov derivative of a polyhedral function enjoys the radial property

$$\dot{v}_{\mathcal{P}}(\mu x) = \mu \dot{v}_{\mathcal{P}}(x) \qquad \forall \mu \geq 0\,, \tag{12}$$

and therefore its sign behaviour in $\mathbb{R}^n$ can be inferred by the behaviour on the boundary of the polyhedron $\mathcal{P}$. Notice that the maximum value of the linear function

$$p_k^T Ax\,, \tag{13}$$

on the k-th face is attained at the vertices of the k-th face. Hence it follows that a necessary and sufficient condition for $Dv_{\mathcal{P}}(x)$ to be negative definite is that, for all $k = 1, \ldots, s$, $p_k^T Ax < 0$ at the vertices of the k-th face.

From this follows the next important result, which represents the first *extreme point* property we consider in the paper.

Fact 2 $Dv_{\mathcal{P}}(x)$ *is negative definite* if and only if *it is strictly negative on the vertices of* $\mathcal{P}$.

By virtue of Fact 2 the last part of this section will be devoted to solve the following problem.

Problem 1 Find an integer m and m points $\{\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(m)}\}$ such that their convex hull induces a polyhedral Lyapunov function for system (1).

Due to the linearity of the system under consideration, we can restrict our attention to symmetric polyhedral functions: it is possible to show that also this class is universal.

For the sake of simplicity we shall consider the case $n = 2$; however, in principle, the procedure is generalizable to the n-dimensional case. Given a set of non-zero points $\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(m)} \in \mathbb{R}^2$ we will assume they are ordered counter-clockwise.

Theorem 1 *A collection of symmetric points $\{\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(m)}\}$ ($m \geq 4$ and even) is a solution of Problem 1 if and only if there exists a set of row vectors $\boldsymbol{p}_j^T \in \mathbb{R}^2$, $j = 1,\ldots,m/2$, such that the following constraints are satisfied for $i = 1,\ldots,m/2-1$,*

$$\begin{aligned} \boldsymbol{p}_i^T A\boldsymbol{x}^{(i)} &< 0 && \text{(14a)}\\ \boldsymbol{p}_i^T A\boldsymbol{x}^{(i+1)} &< 0 && \text{(14b)}\\ \boldsymbol{p}_{m/2}^T A\boldsymbol{x}^{(m/2)} &< 0 && \text{(14c)}\\ -\boldsymbol{p}_{m/2}^T A\boldsymbol{x}^{(1)} &< 0 && \text{(14d)} \end{aligned}$$

$$\begin{aligned} \boldsymbol{p}_i^T \boldsymbol{x}^{(i)} &= 1 && \text{(15a)}\\ \boldsymbol{p}_i^T \boldsymbol{x}^{(i+1)} &= 1 && \text{(15b)}\\ \boldsymbol{p}_{m/2}^T \boldsymbol{x}^{(m/2)} &= 1 && \text{(15c)}\\ -\boldsymbol{p}_{m/2}^T \boldsymbol{x}^{(1)} &= 1 && \text{(15d)} \end{aligned}$$

and, for $m \geq 6$, the following constraints are satisfied for $i = 1,\ldots,m/2-1$, $j = 1,\ldots,m/2$

$$\begin{aligned} \pm\, \boldsymbol{p}_i^T \boldsymbol{x}^{(j)} &< 1 \qquad j \neq i,\, j \neq i+1 && \text{(16a)}\\ \pm \boldsymbol{p}_{m/2}^T \boldsymbol{x}^{(j)} &< 1 \qquad j \neq 1,\, j \neq m/2 && \text{(16b)} \end{aligned}$$

Proof. Constraints (15) follow from the fact that the i-th line intersects the polyhedron in correspondence of the i-th and $(i+1)$-th vertex.

From Fact 1 constraints (14) assure that the derivative is negative definite.

Now we notice that constraints (15) above do not guarantee that the polygon we obtain is actually convex (note that the surface obtained from the intersection of the halfplanes is always polyhedral, but generally it will not have as vertices $\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(m)}$). The convexity constraints can be stated as follows: *given two vertices $\boldsymbol{x}^{(i)}$, $\boldsymbol{x}^{(i+1)}$ and the line described by $\boldsymbol{p}_i^T$ passing through them, all other vertices must be contained in the interior of the i-th halfplane*, from which constraints (16) follow. These are $2(m/2-2)$ constraints for each of the $m/2$ halfplanes. ■

Note that the constraints (15) can be used to express the row vectors $\boldsymbol{p}_i^T$, $i = 1,\ldots,m/2$, as functions of the vertices $\boldsymbol{x}^{(i)}$, $i = 1,\ldots,m/2$; then, substituting in (14) and (16), we reduce our original Problem 1 to find $\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(m/2)}$ so as to satisfy a set of strict inequality constraints. This feasibility problem can be solved using standard optimization routines (see also the Example below).

4 Stability Analysis of Uncertain Systems

In this section we turn our attention to the uncertain system (1).

Definition 2 System (1) is said to be polyhedrally stable if there exist an integer m and a polyhedron $\mathcal{P}$ having m vertices such that the Lyapunov derivative (10) of the polyhedral function (6) induced by $\mathcal{P}$ is negative definite for all $\pi \in \mathcal{R}$.

Polyhedral stability guarantees uniform asymptotic stability and, hence, exponential stability of system (1). Again in this case an extreme point result holds.

Theorem 2 *System (1) is polyhedrally stable* if and only if *there exist an integer m and a polyhedron $\mathcal{P}$ having m vertices such that the derivative (10) of the polyhedral function (6) induced by $\mathcal{P}$ is negative definite on each of the vertices of $\mathcal{R}$.*

Proof. First assume that in (3) the denominator is identically equal to one (that is the system matrix is multiaffine in the parameters). The derivative of (6) along the solutions of system (1) is given, according to (10), by

$$Dv_{\mathcal{P}}(x, \pi) = \max_{j \in I(x)} p_j^T A(\pi) x\,. \tag{17}$$

Now the above derivative is negative for all $\pi \in \mathcal{R}$ if and only if its maximum on $\mathcal{R}$ is negative. We have

$$\begin{aligned} \max_{\pi \in \mathcal{R}} Dv_{\mathcal{P}}(x, \pi) &= \max_{\pi \in \mathcal{R}} \max_{j \in I(x)} p_j^T A(\pi) x \\ &= \max_{j \in I(x)} \max_{\pi \in \mathcal{R}} p_j^T A(\pi) x \end{aligned} \tag{18}$$

The proof follows from the fact that a multiaffine function defined on a hyperrectangle attains both its maximum and its minimum at one of the vertices (see [2]).

Now we turn on the more general situation in which $A(\cdot)$ is the ratio of multiaffine polynomials in π. The proof readily follows by observing that, since the denominator of (3) is a multiaffine function of π, it does not change sign in $\mathcal{R}$ if and only if it does not change sign on the set of the vertices. ∎

Remark 1 For $m = 4$ the constraints (16) need not to be checked.

Remark 2 The general nonlinear case can be approached using the results in [1].

By virtue of Theorem 2 our goal is to solve the following problem.

Problem 2 Find an integer m and a suitable polyhedron $\mathcal{P}$ of m vertices such that the derivative of the induced polyhedral function is negative definite on each of the vertices of $\mathcal{R}$.

Note that, since the polyhedron $\mathcal{P}$ is common to the whole family of uncertain systems, the $m(m/2-2)$ convexity constraints (16) remain unchanged. To impose that the derivative is negative definite on the vertices of $\mathcal{R}$ we have to consider 2^{n_p} set of constraints of the form (14) (one set for each vertex of $\mathcal{R}$).

Example This example is taken from [10]. Consider

$$A(\pi) = \begin{pmatrix} 0 & 1 \\ -1+\pi & -1 \end{pmatrix} \tag{19}$$

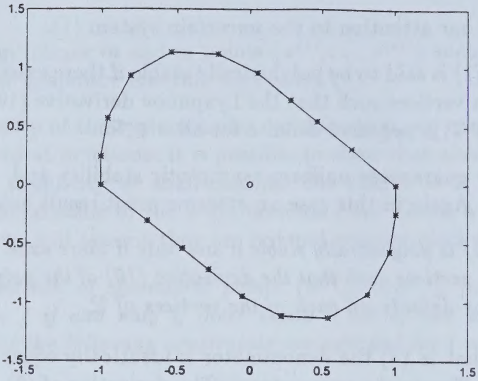


Figure 1:

with $|\pi| \leq \gamma$. To solve Problem 2 we used the Matlab optimization routine *Minimax* with cost function given by

$$\begin{aligned}&\min\max f(x^{(1)},\dots,x^{(m/2)})\\ &:= \min_{i=1,\dots,2^{n_p}} \max\left\{p_1^T A(\pi_{(i)})x^{(1)}, p_1^T A(\pi_{(i)})x^{(2)},\dots,p_{m/2}^T A(\pi_{(i)})x^{(m/2)}, -p_{m/2}^T A(\pi_{(i)})x^{(1)}\right\}\end{aligned} \tag{20}$$

where $\pi_{(i)}$ represents the i-th vertex of $\mathcal{R}$. We established that the system under consideration is polyhedrally stable for $\gamma = 0.999$ using the polyhedral function of 20 vertices depicted in Figure 1. Note that the quadratic stability radius of this system is $\gamma = 0.94$, while, using polyhedral functions according to the method developed in [4], the robustness bound was $\gamma = 0.98$. Only recently in [5] the same bound $\gamma = 0.999$ has been found with a polyhedral function of 58 vertices.

5 Conclusions

In this paper we have considered a linear system depending on a set of uncertain parameters in the form of the ratio of multiaffine polynomials. A procedure which allows to prove the exponential stability of the system under consideration, using the class of polyhedral Lyapunov functions, has been provided. A first contribute is the direct construction of a polyhedral function for a given certain continuous system; this procedure is then used to deal with uncertain systems. Our machinery uses extreme points results. Future research will be devoted to develop a procedure to handle the general n-dimensional case.

References

[1] F. Amato, F. Garofalo, L. Glielmo and A. Pironti, “Robust and quadratic stability via polytopic set covering” *Int. J. Robust and Nonlinear Control*, to appear, 1995.

[2] B. R. Barmish and H. I. Kang, "A survey of extreme point results for robustness of control systems," *Automatica*, vol. 29, pp. 13–35, 1993.

[3] F. Blanchini, "Ultimate boundedness control for uncertain discrete-time systems via set-induced Lyapunov functions," *IEEE Trans. Aut. Contr.*, vol. AC-39, pp. 428–433, 1994.

[4] F. Blanchini, "Non-quadratic Lyapunov functions for robust control," *Automatica*, to appear, 1995.

[5] F. Blanchini, *private communication.*

[6] S. Boyd, L. El Ghaoui, E. Feron and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, SIAM press, 1994.

[7] R. K. Bryton and C. H. Tong, "Stability of dynamical systems: a constructive approach," *IEEE Trans. Circ. Syst.*, vol. CAS-26, pp. 224–234, 1979.

[8] M. Corless, "Robust stability analysis and controller design with quadratic Lyapunov functions," in *Variable Structure and Lyapunov Control*, A. S. I. Zinober ed., Lectures Notes in Control and Information Sciences, Springer-Verlag, 1994.

[9] W. Hahn, *Stability of Motion*, Springer Verlag, Berlin, 1967.

[10] A. Olas, "On robustness of systems with structured uncertainties," *Proc. 4th Workshop on Control Mechanics*, California, 1991.

[11] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1972.

Supervisory Control Synthesis from Metric Temporal Logic Specifications*

M. Barbeau, F. Kabanza, and R. St-Denis
Département de mathématiques et d'informatique
Université de Sherbrooke
Sherbrooke, Québec CANADA J1K 2R1
{barbeau, kabanza, stdenis}@dmi.usherb.ca

Abstract

We have introduced a novel controller synthesis method that handles, in a single framework, the issues of controllability, safety, liveness, and real time. Our approach is founded on the supervisory control theory of Ramadge and Wonham and on Metric Temporal Logic (MTL). The highlights of our method consist of a new technique for incrementally checking MTL formulas over sequences of states generated by occurrences of events and a new method for backtracking during search by taking into account uncontrollable events. The overall approach was introduced in a previously published paper in which, however, we treated only bounded-time formulas. Herein, we generalize the approach to unbounded-time liveness formulas to cover the whole range of MTL formulas.

1 Introduction

A controller is a program that specifies the events that can occur during the execution of a process. The main concern of this paper is automatic synthesis of controllers of discrete-event systems from control requirements expressed as temporal properties.

Temporal logics have been recognized as a useful tool for dealing with temporal properties [Eme90]. Real-time temporal logics, such as Metric Temporal Logic (MTL), can express real time properties of the form "each occurrence of event e leads to satisfaction of condition c within five time units" [AH93]. One interesting development in this area

---

*The research described in this paper was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Fonds pour la formation de chercheurs et l'aide à la recherche (FCAR).

is *model checking* [Eme90]. In this approach, verifying a program is viewed as evaluating the truth of a temporal formula on a temporal structure representing the program. In a previous paper [BKSD95], we introduced a controller synthesis approach for handling a restricted class of MTL formulas called *bounded-time MTL formulas.* The example given above is expressible as a bounded-time formula. In this paper, we extend our algorithm to handle unbounded-time formulas. An example of a constraint expressible as an unbounded-time formula is "each occurrence of event e eventually leads to satisfaction of condition c".

One interesting feature of our technique is that it does not require explicit storage of the state-transition graph over which formulas are checked. Rather, our algorithm generates and checks state sequences *on the fly* (i.e., incrementally). This allows us to generate controllers using a standard forward-chaining search. The search process is based on a technique called *control-directed backtracking*, which uses selective backtracking to states with outgoing controllable paths. This approach allows significant pruning of the search space [BKSD95].

The paper is organized as follows. Section 2 presents the problem and a review of our synthesis algorithm. Section 3 describes our extended algorithm for handling unbounded-time eventualities. Finally, Section 4 offers our conclusions.

2 Review of Previous Work

The controller synthesis algorithm in [BKSD95] takes as input a process modeled as a state-transition system, a set of uncontrollable events, and a legal behavior specified as an MTL formula. Each event is associated with an activity that has a duration expressed as an integral number of time units. The algorithm generates a supervisor by performing a forward-chaining search to explore state sequences generated by the execution of the process. The supervisor is extracted from the state sequences that satisfy the MTL formula, that is, the satisfactory sequences. The algorithm returns a controller for which combination with the process insures that the behavior of the latter remains within the scope of legal behavior.

MTL temporal operators are U (*until*), $\Box$ (*always*), and $\Diamond$ (*eventually*), subscripted with a time constraint of the form "$\leq x$" or "$\geq x$", where x is an integer. If the constraint has the form "$\leq x$" ("$\geq x$"), the modal formula must hold before (after) the delay x expires. Note that the formula $\Diamond_{\leq x}$ ($\Diamond_{\geq x}$) is an abbreviation of the formula $true\, U_{\leq x}\, f_2$ ($true\, U_{\geq x}\, f_2$). Eventualities are formulas for which the main operator is U. An eventuality subscripted with the symbol $\leq$ is said to be bounded-time, whereas one subscripted with the symbol $\geq$ is termed unbounded-time. A bounded-time formula is one in which only propositional symbols are negated and has no unbounded-time eventualities.

Satisfaction of an MTL formula is checked *on the fly*, which means that the storage of a complete representation of the process's state-transition system is not required. The algorithm, in general, contents itself with a partial representation of the state-transition system to allow lower space requirements.

Our mechanism for checking state sequences on the fly hinges on a notion of progressing formulas through states. Given a state, assessment of a formula can be seen as a local evaluation of the formula on that state, which is called the *present requirement*, and the derivation of a *future requirement* to be checked on the subsequences starting from that state. This view is valid with other modal temporal logics, including MTL. More formally, we defined a progression function that takes as input an MTL formula f, a state s, an elapsed period of time t, and a function π that evaluates propositional symbols in s (function $\pi(p, s)$ returns true if p holds in s). The progression function assesses f in s. If the present requirement is violated in s, $false$ is returned. If the present requirement is satisfied and if there is no future requirement to satisfy, then $true$ is returned. Otherwise, an MTL formula f' expressing the future requirement is returned.

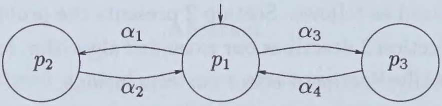


Figure 1: The process

Let us consider a process modeled as the state-transition system of Figure 1. Every event is associated with an activity that has one time unit duration. There are three states, namely, $\{p_1\}$, $\{p_2\}$, and $\{p_3\}$. The evaluation function $\pi(p, s)$ returns true if $p \in s$. The legal behavior is specified as the bounded-time MTL formula:

$$g \equiv \Box_{\geq 0}(p_1 \rightarrow \Diamond_{\leq 3}\, p_2) \land \Box_{\geq 0}(p_1 \rightarrow \Diamond_{\leq 3}\, p_3)$$

The formula g states that, in every state, if p_1 holds, then p_2 and p_3 must have been satisfied before three units of time have elapsed, but not necessarily at the same moment.

Now, let us consider the progression of g in state $\{p_1\}$ in Figure 1. The present requirement is true. Since $\pi(p_1, \{p_1\})$ is true, progression of g in state $\{p_1\}$ with a null elapsed period of time yields the following future requirement:

$$f_1 \equiv g \land \Diamond_{\leq 3}\, p_2 \land \Diamond_{\leq 3}\, p_3$$

This states that, from state $\{p_1\}$, formula g must be satisfied and p_2 and p_3 must have been satisfied before three units of time have elapsed. Note that g must always be satisfied from one state to another because its main operator is $\Box_{\geq 0}$.

Now, let us consider the progression with one unit of time of f_1 in state $\{p_2\}$. The result is:

$$f_2 \equiv g \wedge \Diamond_{\leq 2} p_3$$

Since $\pi(p_2, \{p_2\})$ is true, the eventual satisfaction of p_2 is met and erased from the future requirement. Time has progressed one unit and p_3 is not satisfied. Proposition p_3 must still be satisfied before two units of time have elapsed; this appears in f_2 as $\Diamond_{\leq 2} p_3$.

Finally, the progression with three units of time of f_1 in state $\{p_2\}$ would yield *false*, because the limits allowed for satisfying p_3 have expired. The formula progression algorithm is Algorithm 2.1.

Algorithm 2.1

```
Prog($f$,$s$,$t$,$\pi$)
1. Case $f$
2.    $p$ ($p$ a proposition): $\pi(p, s)$;
3.    $\neg f_1$ : $\neg Prog(f_1, s, t, \pi)$;
4.    $f_1 \wedge f_2$: $Prog(f_1, s, t, \pi) \wedge Prog(f_2, s, t, \pi)$;
5.    $f_1 \vee f_2$: $Prog(f_1, s, t, \pi) \vee Prog(f_2, s, t, \pi)$;
6.    $\Box_{\leq x} f_1$: Case $x - t > 0$: $Prog(f_1, s, 0, \pi) \wedge \Box_{\leq (x-t)} f_1$
                   $x - t = 0$: $Prog(f_1, s, 0, \pi)$;
                   otherwise: true;
7     $\Box_{\geq x} f_1$: Case $x - t > 0$: $\Box_{\geq (x-t)} f_1$
                   otherwise: $Prog(f_1, s, 0, \pi) \wedge \Box_{\geq 0} f_1$;
8.    $f_1 \, U_{\leq x} \, f_2$: Case $x - t > 0$: $Prog(f_2, s, 0, \pi) \vee$
                          $(Prog(f_1, s, 0, \pi) \wedge f_1 \, U_{\leq (x-t)} \, f_2)$
                   $x - t = 0$ $Prog(f_2, s, 0, \pi)$;
                   otherwise: false;
9.    $f_1 \, U_{\geq x} \, f_2$: Case $x - t > 0$: $f_1 \, U_{\geq (x-t)} \, f_2$
                   otherwise: $Prog(f_2, s, 0, \pi) \vee$
                          $(Prog(f_1, s, 0, \pi) \wedge f_1 \, U_{\geq 0} \, f_2)$
```

■

One thus generates a supervisor by searching for satisfactory sequences of states, using formula progression. Every state of the supervisor results from a state s of the process paired with its future requirement f. This is denoted as $< s, f >$. One starts from the initial state, labeled with the progression, with zero units of time, of the input MTL formula. Given a pair $< s, f >$ for each transition (s, α, s') in the process, if f' is the progression of f in s, with the duration of the activity associated with the event α, then one generates a successor $< s', f' >$ and a transition $(< s, f >, \alpha, < s', f' >)$. Every state labeled with the formula *false* is a *sink* (a dead end). A state from which there is an uncontrollable transition to a sink is also a sink. Thus, we have nontrivial sinks that consist of states from which there are paths of uncontrollable transitions to basic sinks (that is, states labeled with false). Control-directed backtracking consists in taking into account this notion on nontrivial sink to backtrack to the most recent non-sink state when a basic sink is encountered. This can lead to significant savings [BKSD95].

This search process is characterized by the fact that, for formulas without unbounded-time eventualities, each path terminated by a cycle, without any state labeled by the formula *false*, is satisfactory. In other words, it satisfies the formula labeling the initial state on the path. The search process generates sequences satisfying the input MTL formula. The supervisor feedback is generated by prohibiting transitions from satisfactory states towards sinks.

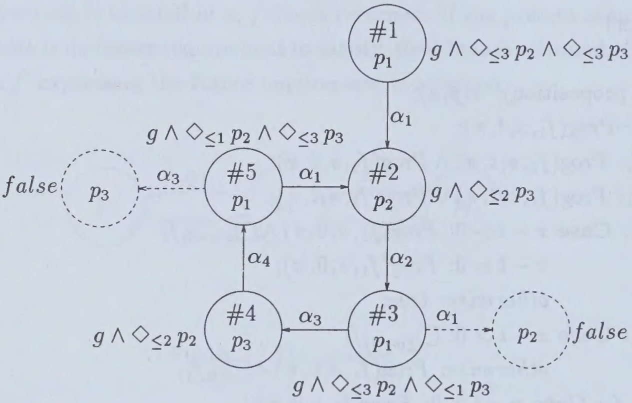

#1
p_1
$g \wedge \Diamond_{\leq 3} p_2 \wedge \Diamond_{\leq 3} p_3$
α_1
$g \wedge \Diamond_{\leq 1} p_2 \wedge \Diamond_{\leq 3} p_3$
false
p_3
α_3
#5
p_1
α_1
#2
p_2
$g \wedge \Diamond_{\leq 2} p_3$
α_4
α_2
$g \wedge \Diamond_{\leq 2} p_2$
#4
p_3
α_3
#3
p_1
α_1
p_2
false
$g \wedge \Diamond_{\leq 3} p_2 \wedge \Diamond_{\leq 1} p_3$

Figure 2: The state transition system of the supervisor

Figure 2 pictures a state-transition diagram of a supervisor synthesized for the process in Figure 1, given the requirements expressed as the MTL formula g. Dotted arrows and circles correspond to transitions and states inspected during the search process but rejected as unsatisfactory. If we assume that both events α_1 and α_3 are controllable, then the supervisor feedback function must prohibit them in states #3 and #5, respectively, because they lead to sinks.

This search process alone does not, however, address unbounded-time eventualities. The problem with such an eventuality is that, when it is not satisfied in the current state, its progression to a successor yields exactly the same eventuality: the time constraint "≥ 0" is invariant. Hence, the formula labeling a given state can be identical to that labeling an ancestor state while not achieving an unbounded-time eventuality in this formula. In other words, cycles may be unsatisfactory, that is, correspond to infinite sequences not satisfying the formula labeling the initial state. Let us consider an example.

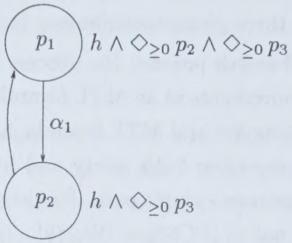

p_1
$h \wedge \Diamond_{\geq 0} p_2 \wedge \Diamond_{\geq 0} p_3$
α_2
α_1
p_2
$h \wedge \Diamond_{\geq 0} p_3$

Figure 3: A cycle

Let us define the MTL constraint:

$$h \equiv \Box_{\geq 0}(p_1 \rightarrow \Diamond_{\geq 0} p_2) \wedge \Box_{\geq 0}(p_1 \rightarrow \Diamond_{\geq 0} p_3)$$

interpreted as whenever p_1 holds, then both p_2 and p_3 must eventually hold, but not necessarily at the same moment. Application of the previous search algorithm produces a supervisor state-transition structure that includes the loop pictured in Figure 3. This is of course wrong because the supervisor may loop infinitely on states $< \{p_1\}, h \wedge \Diamond_{\geq 0} p_2 \wedge \Diamond_{\geq 0} p_3 >$ and $< \{p_2\}, h \wedge \Diamond_{\geq 0} p_3 >$ without ever satisfying $\Diamond_{\geq 0} p_3$. An extended search process handling both bounded and unbounded-time eventualities is discussed in the next section.

3 Search Process Algorithm

The extended search process is comparable to techniques for synthesizing process synchronizers from temporal logic formulas without time constraints [EC82] [Wol89].

In both [EC82] and [Wol89], the process behavior and required properties are specified as a single temporal formula without time requirements.[1] In [EC82], a graph is generated from the formula. Execution sequences of this graph satisfy safety requirements. Afterwards, satisfaction of liveness requirements is assessed on the produced graph (as a matter of fact, bounded-time formulas express safety properties). States and transitions of the graph that satisfy the liveness requirements define the state-transition structure of the process synchronizer. This technique, therefore, requires two phases.

In [Wol89], a process synchronizer is obtained by constructing a Büchi automaton that accepts infinite sequences satisfying the given formula. This is done by constructing a *local automaton*, accepting sequences satisfying the safety requirements, and an *eventuality automaton*, accepting sequences satisfying the liveness requirements. The Büchi automaton for the entire formula is then obtained as a product of the two automata. This is therefore a three-phase technique.

---

[1] Note that [EC82] addresses branching-time temporal logic, while [Wol89] is a survey of approaches addressing both branching and linear-time temporal logics

In our extended search process, the process behavior is specified as a state-transition system and the requirements as an MTL formula; that is, requirements refer to real time. From the process behavior and MTL formula, a state-transition system for the controller is generated while assessing both safety and liveness requirements *on the fly*. In other words, our technique uses a single pass. Furthermore, we take into account controllability of events, which is not in [EC82] or [Wol89].

Our extended search process is described in more detail hereafter. In addition to the MTL formula, each state is labeled with a set of unbounded-time eventualities to achieve. This set corresponds to a state of the eventuality automaton in [Wol89]. Hence, sets of unbounded-time eventualities are progressed in accordance to the rules defining transitions in the construction process of the eventuality automaton. Sets of eventualities are progressed in order to check that, sooner or later, they are made true. Intuitively, we must ensure that whenever a formula of the form $f_1 \, U_{\geq x} \, f_2$ labels a state, there is descendant satisfying f_2 for each sequence outgoing from this state. Satisfactory sequences, with respect to safety and liveness requirements, are therefore checked on the fly.

Special attention must be paid to disjunctive formulas. Given a requirement expressed as a formula of the form $f_1 \vee f_2$, if f_1 holds, then f_2 is not required, and vice versa. Each requirement is decomposed into disjunctive normal form. For each disjunct, which is a conjunction, a nondeterministic successor is generated. Hence, every state is labeled only with a conjunction of formulas.

In this way, f_2 must be satisfied on some state on the cycle for each unbounded-time eventuality $f_1 \, U_{\geq x} \, f_2$ which is a conjunct in the labelling of the starting state.

The extended search process generates a supervisor for which every state results from a state s of the process, a future requirement f, and a set of unbounded-time eventualities to achieve E. This is written as $< s, f, E >$. There are as many initial states as there are disjuncts in the normal form of the formula representing the initial requirement. Every initial state is constructed from the initial state of the process; a disjunct resulting from the progression, with zero units of time, of the input MTL formula; and the set of unbounded-time eventualities occuring as conjuncts in this disjunct. Given a supervisor state $< s, f, E >$ for each transition (s, α, s') in the process, a successor $< s', f', E' >$ and a transition $(< s, f, E >, \alpha, < s', f', E' >)$ are generated where f' is the progression of f in s, with the duration of the activity associated to the event α. The set E' is computed by Algorithm 3.1 as follows:

Algorithm 3.1 [Computation of E' from E]

1. if $E = \phi$, then E' is the set of unbounded-time eventualities $f_1 \, U_{\geq x} \, f_2$ appearing as conjuncts in f' such that f_2 is not a conjunct in f',

2. else E' is obtained from E by removing every eventuality $f_1\, U_{\geq x}\, f_2$ such that f_2 is a conjunct in f' or f_2 is propositional and $\pi(f_2, s)$ is true.

■

A cycle is considered satisfactory if it does not contain a sink and has a state for which the set of unbounded-time eventualities is empty. In other words, the computed search graph is a Büchi automaton accepting all sequences of the process satisfying the input MTL formula. The accepting states are those labeled with the empty set of unbounded-time eventualities to achieve. The extended search process is described more formally as follows:

Algorithm 3.2 [Search Process]
Input: an MTL formula f, a process state-transition system (with initial state s_0), and a set of uncontrollable events.
Output: a supervisor Büchi automaton.

1. Let f' be the progression of f into s_0 with time difference 0.
2. Transform f' into disjunctive normal form using De Morgan and logical equivalences; i.e., $f' \equiv g_1 \vee \ldots \vee g_n$, with $g_i \equiv f_{i1} \wedge \ldots \wedge f_{in_i}$.
3. For each $g_i \neq \mathit{false}$, create an initial state $< s_0, g_i, E >$, with E the set of formulas f_{ij} of the form $f_1\, U_{\geq x}\, f_2$ such that f_2 is not a conjunct in g_i.[2]
4. Repeatedly select a current state $< s, f, E >$ and expand it as follows. For each transition (s, α, s') in the process, let f' be the progression of f in s', with $f' \equiv g_1 \vee \ldots \vee g_n$ the disjunctive normal form of f' and for each g_i,
    (a) generate a successor state $< s', g_i, E' >$, where E' is obtained from E following Algorithm 3.1.
    (b) generate the transition $(< s, f, E >, \alpha, < s', g_i, E' >)$.

■

The decomposition of formulas into disjunctive normal form introduces nondeterministic transitions. An event enabled in a state is satisfactory if one of the transitions labeled by the event leads to a satisfactory state. In other words, we have a nondeterministic Büchi automaton accepting all sequences of the process satisfying the formula. One can build a corresponding deterministic Büchi automaton by using the technique in [Saf88].

As an example, a Büchi automaton representing the supervisor generated for the process pictured in Figure 1 and requirements expressed as the formula h is illustrated

---

[2] We thus have many initial states. One can, however, reduce this to a single initial state with nondeterministic ϵ-transitions to these states.

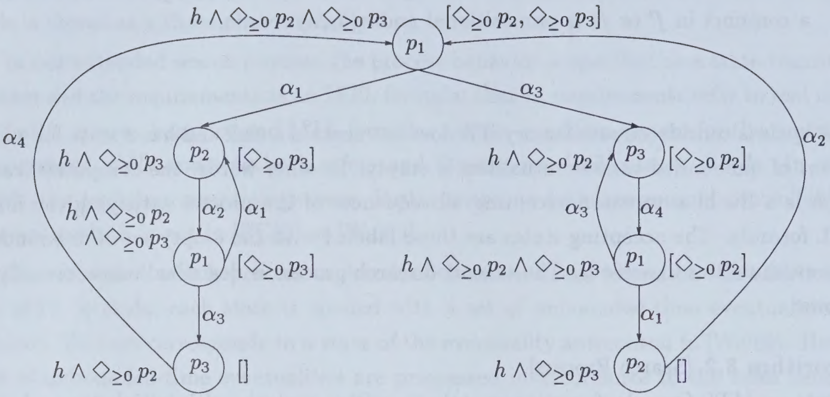

$h \wedge \Diamond_{\geq 0} p_2 \wedge \Diamond_{\geq 0} p_3$
$[\Diamond_{\geq 0} p_2, \Diamond_{\geq 0} p_3]$
p_1
α_1
α_3
α_4
$h \wedge \Diamond_{\geq 0} p_3$
p_2
$[\Diamond_{\geq 0} p_3]$
$h \wedge \Diamond_{\geq 0} p_2$
p_3
$[\Diamond_{\geq 0} p_2]$
α_2
$h \wedge \Diamond_{\geq 0} p_2$
$\wedge \Diamond_{\geq 0} p_3$
α_2
α_1
α_3
α_4
p_1
$[\Diamond_{\geq 0} p_3]$
$h \wedge \Diamond_{\geq 0} p_2 \wedge \Diamond_{\geq 0} p_3$
p_1
$[\Diamond_{\geq 0} p_2]$
α_3
α_1
$h \wedge \Diamond_{\geq 0} p_2$
p_3
$[]$
$h \wedge \Diamond_{\geq 0} p_3$
p_2
$[]$

Figure 4: The search graph

in Figure 4. The future requirement MTL formula is placed on the left of each state and the set of eventualities on the right.

As our algorithm stands, we need to deal with the controllability of infinite behaviors, which are modeled as cycles in the Büchi automaton. To illustrate this, let us consider the transition in Figure 4 labeled with event α_1 from state $< \{p_1\}, h \wedge \Diamond_{\geq 0} p_2 \wedge \Diamond_{\geq 0} p_3, [\Diamond_{\geq 0} p_3] >$ to state $< \{p_2\}, h \wedge \Diamond_{\geq 0} p_3, [\Diamond_{\geq 0} p_3] >$. This transition completes a loop that does not contain an accepting state. If α_1 is uncontrollable, a necessary eventual escape from the loop may not be insured by the controller. Hence, if both events α_1 and α_3 are uncontrollable, there is no controller solution given the process pictured in Figure 1 and the MTL formula h. Handling uncontrollable loops that do not contain accepting states needs further research. The controllability of infinite behaviors is also addressed in [TW94].

4 Conclusion

We have presented a synthesis algorithm handling both bounded and unbounded real time properties. A legal behavior is specified as an MTL formula. The acceptance condition of a Büchi automaton is that an infinite sequence is accepted if it contains an accepting state (that is, a state containing the empty set of eventualities in our case) infinitely often. This characterizes all different legal implementations of the supervisor. One can unwind a specific supervisor from the Büchi automaton by observing that each strongly connected component containing an accepting state can be unwinded into accepting infinite sequences. One must, however, take into account the fact that some transitions are uncontrollable. Roughly speaking, the supervisor inhibits controllable transitions that lead to sinks or cause cycles that do not involve an accepting state. Note that a specific supervisor can be constructed on the fly by slightly modifying the previous algorithm to detect sinks and cycles that do not involve accepting states.

References

[AH93] R. Alur and T. Henzinger. Real-time logics: Complexity and expressiveness. *Information and Computation*, 104(1):35–77, 1993.

[BKSD95] M. Barbeau, F. Kabanza, and R. St-Denis. Synthesizing plant controllers using real-time goals. In *Proc. of 14th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 791–798, 1995.

[EC82] E. A. Emerson and E. M. Clarke. Using branching time logic to synthesize synchronization skeletons. *Science of Computer Programming*, 2(3):241–266, 1982.

[Eme90] E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 995–1072. MIT Press/Elsevier, 1990.

[MW84] Z. Manna and P. Wolper. Synthesis of communicating processes from temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 6(1):68–93, 1984.

[Saf88] S. Safra. On the complexity of omega-automata. In *Proc. of the 29th IEEE Symposium on Foundations of Computer Science*, White Plain, October 1988.

[TW94] J. G. Thistle and W. M. Wonham. Supervision of infinite behaviors of discrete-event systems. *SIAM Journal of Control and Optimization*, 32(4):1098–1113, 1994.

[Wol89] P. Wolper. On the relation of programs and computations to models of temporal logic. In *Proc. Temporal Logic in Specification*, pages 75–123. Lecture Notes in Computer Science, Vol. 398, 1989.

Strong Consistency of Infinitesimal Perturbation Analysis for Symmetric ATM Intree Networks

Charles A. Brooks[1]
Electrical Engineering and Computer Science Department
University of Illinois at Chicago
cbrooks@eecs.uic.edu

Pravin Varaiya
Electrical Engineering and Computer Science Department
University of California, Berkeley

Abstract

Discrete Event System Infinitesimal Perturbation Analysis (IPA) can be used to determine estimates of the gradients of the average delay with respect to each link capacity in a computer communications network. It is shown that an IPA gradient estimate can provide unbiased and strong consistent gradient estimates for a class of ATM Intree networks with symmetric traffic. Leaky-bucket input regulation and independent Poisson arrival processes are assumed. A Generalized Semi-Markov Process (GSMP) model is presented in the analysis.

1 Introduction

Perturbation Analysis (PA) is an efficient method for the computation of gradients from a single (real or simulated) sample path. Through the use of a simple IPA algorithm, network planners can use gradient-based optimization methods for determining link capacities and network performance prior to physically implementing the network. The advantage of IPA is that it avoids the usage of many simplifying assumptions generally used by network planners to optimize performance of their networks.

The strongest assumption in use is the "independence assumption." It is assumed that all packet service times are independent and exponentially distributed. Therefore a packets service time at a particular link is independent of its service time at any preceding or succeeding link. However in real networks a packet that traverses two network links that have identical capacities will experience the same service time on each link, and as a result service times will be correlated.

The independence assumption has been used in much of the past and present network planning literature (e.g., in [7, 8]). Successive work avoided the use of the independence assumption by employing Perturbation Analysis. In [1] simulation-based IPA is used to determine routing and capacity assignment, and in [11] the use of IPA for dynamic bandwidth allocation is demonstrated. In both cases simulation results indicate that IPA provides increased performance accuracy over traditional methods.

However the benefits of these algorithms to network planners are somewhat subjective due to biases in the gradient estimates introduced by event order changes and nondifferentiabilities. To show that perturbation analysis yields the desirable results, it is

---

[1] Research supported by the AT&T Foundation through the AT&T Cooperative Research Fellowship Program.

necessary to show that the gradient estimate is unbiased and strong consistent, and that the performance function is differentiable.

It has been shown that ingenuous applications of IPA may yield biased gradient estimates (even with the wrong sign) if there are changes in the order of events between the nominal and perturbed sample paths. In the remainder of this paper, the issues of unbiasedness and strong consistency are addressed for an IPA method for simultaneously determining gradients of the average delay with respect to each link capacity. Differentiability (almost everywhere) of the average delay is also established (via regenerative GSMPs) although it has been shown in [2] that the average delay may have nondifferentiabilities.

Section 2 of this paper presents the network model used in the analysis. The IPA algorithm is discussed in Section 3. Section 4 introduces a GSMP model for the network, and discusses conditions under which it is regenerative. The proof of strong consistency is presented in Section 5. The final section summarizes the results and provides suggestions for future work.

2 ATM Intree Networks

To obtain unbiased and consistent gradient estimates, it is assumed that the communication network has deterministic packet (cell) sizes and an intree topology. Intree networks are a class of acyclic networks in which "overtaking" is not possible. That is, if a cells service precedes the service of a second cell on a link, the service of the second cell must not precede the service of the first cell on a subsequent link.

The topology of the network can be viewed as a connected graph where unidirectional links can fork, merge, or be put in series. At all merges the cell streams are independent. Without loss of generality assume that cell sizes are normalized to one.

An additional assumption is that there is regulation of the external network input to each virtual circuit. For simplicity it is assumed that "leaky buckets" are utilized. The (σ, ρ) notation [3] is used to represent the size of the token bucket and the input rate of tokens. In section 5.1, it is shown that under certain stability conditions, usage of input regulation with finite token bucket sizes results in finite bounds for queue backlogs.

Note that it is assumed that ATM switches in the network use output queueing, and all delays in the switch are negligible with respect to the delay experienced at the output link. Also note that the time spent in a regulator does not contribute to the cells end-to-end delay.

The final and strongest assumption is that of symmetric traffic: a multi-level star configuration is the basis for the topology, and at each level of the star, all source-destination pairs (or equivalently, virtual circuits) have identical Poissonian (external) traffic arrival rates. This assumption is necessary for unbiasedness. Without loss of generality consider the case of a network having a double-star topology. (See Figure 1.)

3 Infinitesimal Perturbation Analysis

The queueing delay faced by a cell at the output link of an ATM switch is modeled as that in a G/G/1 queue. The basis for the IPA algorithm used for this analysis is presented in detail in [9]. These results are extended to simultaneously estimate the gradient estimates for all links in a network based on a single Monte-Carlo simulation.

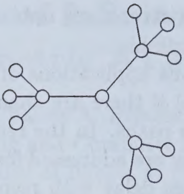


Figure 1: Double-star network

To estimate the gradient of the average delay with respect to the inverse of the link capacity, accumulators measure sample gradients at each link. Note that M sample gradients are computed simultaneously, where M is the number of links in the network. The following procedure is used:

1. Let $i =$ the link at which the capacity is perturbed, and $j =$ the link at which the cell has arrived. All link accumulators are initialized to zero. I.e., $acc_{ij} = 0.0, \forall i, j$.
2. At the beginning of each busy cycle, the accumulator of link j is set to zero if the first arrival in the busy period is an external arrival (i.e., $acc_{ij} = 0.0, \forall i$); or the accumulator of link j is set to the accumulator value of the upstream link, k, from which the cell arrived (i.e., $acc_{ij} = acc_{ik}, \forall i$).
3. For each cell arrival (in the same busy period) at link j, the accumulator measuring sample derivatives with respect to perturbations of link capacity j is incremented by an amount equal to the cell size, which is the derivative of the cell service time with respect to the inverse of the link capacity. Without loss of generality it is assumed that cell sizes (as well as link capacities) are in units of one cell (and cells-per-sec). Therefore the accumulator is incremented by one. I.e., $acc_{jj} = acc_{jj} + 1$.
4. These cell sample derivatives are collected as cells depart the network. The sample derivatives for each link are averaged to determine the gradient estimate for that link.

4 Generalized Semi-Markov Process Model

A Generalized Semi-Markov Process (GSMP) [10, 5] is characterized by the following sextuple

$$(S, A, \xi(\cdot), p(\cdot; \cdot, \cdot), p_0(\cdot), F_{(\cdot)}(\cdot)),$$

where:

S is a state space (finite or countable) representing the set of physical states of a system.

A is a finite set of events.

$\xi(s)$ is the set of possible events in state s ($\xi(s) \subseteq A$).

$p(s'; s, \alpha')$ is the probability of jumping to state s' when event α' occurs in state s.

$p_0(s)$ is the initial state distribution.

$\{F_{\alpha'}(t), \alpha' \in A\}$ is the set of distributions (one for each $\alpha' \in A$) on the non-negative real line used to create new clock samples for events.

4.1 Network GSMP Model

The following GSMP model is used:

The State There are N regulators (leaky buckets) and M queues. For simplicity consider that at each regulator, $\sigma = 1$, and ρ is less than the service time, which implies that the regulators function as queues with deterministic service times. Therefore, the state space of the GSMP is the collection of the state of $N + M$ queues. The state can be written as an (N +M)-tuple, $S = (S_1, \ldots, S_{N+M})$, where S_i is the state of queue i and of the form $S_i = I$ or $S_i \in \bigcup_{l \in Z_+} \{d_{i,j} | 1 \leq j \leq l\}$, $d_{i,j} \in \{1, \ldots, L\}$. $S_i = I$ means queue i is empty or idle. Otherwise there are l cells in the queue, with $d_{i,j}$ being the virtual circuit number of the jth cell (L is the maximum virtual circuit number).

The Events $A = \{\alpha \cup \{\beta_i : i \in \{1, \ldots, N + M\}\}\}$, where α = an external network arrival, and β_i = a departure of a cell from link i. Note that at each regulator there is a Poisson process modeling the input. These can be merged into a single aggregate Poisson process whose rate is the sum of all of the individual rates. Therefore there is a single arrival event, α, corresponding to an arrival of the aggregate process. This is also called uniformization or the Standard Clock Algorithm. We assume that two or more events can not occur simultaneously.

The Possible Event List Let S_i = the state of the ith queue. $\alpha \in \xi(s), \forall s \in S$, and $\beta_i \in \xi(s)$, for $s \in \{s : S_i \neq I\}$. For this model note that once a clock is enabled it remains on the possible event list until it expires. This condition is called non-interruptibility.

The Transition Probabilities $p(s'; s, \alpha)$ is a function of the virtual circuit arrival rates, and $p(s'; s, \beta_i)$ is based on the virtual circuit routing table at link i.

The Initial State $p_0(\{\{I\}^{N+M}\}) = 1$. (Initially the network is empty.)

The Clock Distributions $F_\alpha = 1 - \exp(-\lambda t)$ and $F_{\beta_i} = u(\theta_i)$, where λ is the aggregate arrival rate, and $u(\cdot)$ is the unit step function.

4.2 Regeneration

Essential to the proof in Section 5 is the property that this GSMP model is regenerative. In general, a regenerative process, by definition, is a process that continually renews itself. It has the property of forgetfulness, meaning that (at the regeneration points) the future of the process is independent of the past of the process.

4.3 One-Clock State

Let Z_t be the state of the previous ATM intree network GSMP. To prove regeneration the following proposition is needed.

Proposition 4.1 ([4], page 192) *If there is a one-clock state s_0 to which $\{Z_t, t \geq 0\}$ returns infinitely often with probability one, then $\{Z_t, t \geq 0\}$ is regenerative.*

A one-clock state [4, page 190] is defined as state in which there is only one clock enabled. When this clock expires all of the active clocks are reset. Therefore the GSMP regenerates.

Note that the GSMP we are considering regenerates each time it exits state s_0, where $s_0 = \{\{I\}^{N+M}\}$, the empty network, and the one clock that is enabled is α, the arrival event. Poisson arrivals and network stability ensure that state s_0 is visited infinitely often with probability one. Also note that for $\{Z_t, t \geq 0\}$ with one-clock state s_0, if the clock at s_0 has a continuous distribution, then the distribution of cycle lengths, F_c, is non-lattice. As an initial assumption, only Poisson arrivals are considered so (regenerative) cycle times are non-lattice.

4.4 Regenerative GSMPs

The following propositions hold for regenerative GSMPs.

Proposition 4.2 ([4], page 188) *Let $\pi_t(s) = Prob(Z_t = s)$, and $\pi =$ the steady-state distribution of π_t. Let $m_c =$ the common expectation of all cycle lengths, and let $f(s) =$ the sum total of the number of cells in each queue. If $m_c < \infty$, then the following hold:*

(i) *If F_c is non-lattice, $\exists \pi$ such that $\pi_t \Longrightarrow \pi$ as $t \longrightarrow \infty$; where $\Longrightarrow$ corresponds to convergence* in distribution; *hence, there is a Z_∞ such that $Z_t \Longrightarrow Z_\infty$, the steady-state distribution;*

(ii) $\pi(s) = \lim_{t \longrightarrow \infty} \frac{1}{t} \int_0^t 1\{Z_u = s\}\, du$ *a.s.*

Proposition 4.3 ([4], page 188) *Let Ψ_k correspond to the kth exit from one-clock state s_0. If $m_c < \infty$, and F_c is non-lattice,*

$$\pi(s) = \frac{1}{m_c} \mathbf{E}\left[\int_{\Psi_{k-1}}^{\Psi_k} 1\{Z_t = s\}\, dt\right],$$

$\forall k = 1, 2, \ldots$. *Moreover, if $\sum_{s \in S} |f(s)|\, \pi_s < \infty$ then*

$$\lim_{t \longrightarrow \infty} \frac{1}{t} \int_0^t f(Z_u)\, du = \frac{1}{m_c} \mathbf{E}\left[\int_{\Psi_{k-1}}^{\Psi_k} f(Z_t)\, dt\right] \; a.s.$$

Let $Y_n =$ the state of the GSMP after n events, let $t_n =$ the vector of event clock values after n events, and let $\Delta_n = d\tau_{n+1}/d\theta - d\tau_n/d\theta$, with $\tau_n =$ the time of the nth event. Note that $Y_n = Z_{\tau_n}$.

Proposition 4.4 ([4], page 192) *Let K_0 be the index of the first time $Y_n = s_0$. Let K_1 be the index of the second time $Y_n = s_0$. Define $\hat{f}(Y_n, t_n, \Delta_n) \doteq f(Y_n)\Delta_n$. Note that $f(\cdot)$ is a function of the state of the GSMP. For our purposes, $f(Y_n)$ equals the number of cells in the network after 'n' events. If $m_c < \infty$ and*

$$\mathbf{E}\left[\sum_{n=K_0}^{K_1} \left|\hat{f}(Y_n, t_n, \Delta_n)\right|\right] < \infty,$$

then

$$\lim_{N \longrightarrow \infty} \frac{1}{N} \sum_{n=0}^{N-1} \hat{f}(Y_n, t_n, \Delta_n) = \frac{1}{m_c} \mathbf{E}\left[\sum_{n=K_0}^{K_1} \hat{f}(Y_n, t_n, \Delta_n)\right] \; a.s.$$

4.5 Harris Recurrence

Definition 4.1 *A subset of S, S', is recurrent if it can be reached from any state in S in finitely many transitions with probability one. If S' is recurrent, then S' is a regeneration set. A Markov chain with a regeneration set is called Harris recurrent.*

This leads to the following proposition:

Proposition 4.5 ([4]) *If GSMP $\{Z_t, t \geq 0\}$ is recurrent, then (Y_n, t_n, Δ_n) is a Harris recurrent Markov chain; and if $\lim_{n\longrightarrow\infty} \tau_n'(\theta)/n = 0$, (where $\tau_n'(\theta) = d\tau_n/d\theta$), then*

$$\lim_{T\longrightarrow\infty} \frac{L_T'(\theta)}{T} = \lim_{n\longrightarrow\infty} \frac{d}{d\theta}\left[\frac{L_n(\theta)}{\tau_n(\theta)}\right] a.s.,$$

where

$$L_T(\theta) = \int_0^T f(Z_t(\theta))dt, \text{ and } L_n(\theta) = \int_0^{\tau_n} f(Z_t(\theta))dt.$$

5 Strong Consistency

To show strong consistency the following theorem is used. Let

$$\{G_n(\theta), n \geq 0, \theta \in \Theta\}$$

be a family of sequences of sample performance values, and

$$\{g_n(\theta), n \geq 0, \theta \in \Theta\}$$

be a family of sequences of derivative estimates. Suppose $\Theta = [a, b]$, where $0 < a < b$.

Theorem 5.1 ([6, 4]) *Suppose the following hold for all $\theta \in \Theta$.*

(i) $\mathbf{E}[g_n(\theta)] = d\mathbf{E}[G_n(\theta)]/d\theta + b_n(\theta)$, *where* $b_n(\theta) \longrightarrow 0$ *as* $n \longrightarrow \infty$;

(ii) $\lim_{n\to\infty} \mathbf{E}[G_n(\theta)] = \lim_{n\to\infty} G_n(\theta) = l(\theta)$, *a.s., for some* $l(\theta)$;

(iii) $\lim_{n\to\infty} \mathbf{E}[g_n(\theta)] = \lim_{n\to\infty} g_n(\theta) = \gamma(\theta)$, *a.s., for some* $\gamma(\theta)$;

(iv) *the functions* $\{|\mathbf{E}[g_n(\cdot)] - b_n(\cdot)|, n \geq 0\}$ *are bounded by a (deterministic) function* $\overline{g}(\cdot)$ *which is integrable over* Θ.

Then l is differentiable and $\{g_n(\theta), n \geq 0, \theta \in \Theta\}$ is a strongly consistent estimator of $l'(\theta)$, a.e.; that is for almost every $\theta \in \Theta$,

$$\lim_{n\longrightarrow\infty} g_n(\theta) = l'(\theta)\, a.s.$$

Therefore a sufficient proof for strong consistency of the gradient estimates and differentiability of the performance function is to meet these four conditions.

5.1 Input Regulation Results

Since we have independent Poisson external arrivals, the sojourn times of the external queues (regulators) are bounded by M/D/1 queues. If the network is stable, the internal queue i can be modeled as a G/D/1/K$_i$ queue, where K$_i$ is dependent on σ and the inner and outer degrees of the double star topology. Let $d_i =$ the inner degree of the double star topology and let $d_o =$ the outer degree of the double star topology. $K_i = (d_o * d_i - 1)\sigma$ for the outer queues and $K_i = d_o^2(d_i - 1)\sigma$ for the inner queues, assuming that there is a single virtual circuit for each source-destination pair. Let $F = d_i d_o(d_i d_o - 1)\sigma + d_i d_o(d_i d_o - d_o)\sigma = d_i d_o(2d_i d_o - d_o - 1)\sigma =$ the total number of (internal) buffers in the ATM network. For Figure 1, $d_o = 3, d_i = 3$, and $F = 126\sigma$.

The following Lemma ensures that the first and second moments of the (regenerative) cycle time are finite. Let $B' =$ the "aggregate busy period" of the parallel M/D/1 queues, and let $\lambda' =$ the arrival rate of each virtual circuit.

Lemma 5.1 *If* $\lambda' < \rho$, *then* $\{Z_t, t \geq 0\}$ *is regenerative with cycle time c, such that* $\mathbf{E}(c^2) < \infty$.

Proof: The GSMP has a one-clock state so it is regenerative. Each cycle consists of two components, a busy period and an idle period, each of which are independent and identically distributed. The busy and idle periods are also independent of each other.

The finite buffer queues will empty in a finite amount of time, so the busy period is dominated by identical, independent M/D/1 queues in parallel. The idle period is exponentially distributed with a rate equal to the sum of the individual service rates. Therefore the major concern is the duration of the busy period for the parallel M/D/1 queues. It is known that for a stable M/D/1 queue, the first and second moments of its busy period are finite. Therefore by considering B' to be a geometric distribution of the single queue busy period, B' has finite first and second moments.

5.2 Condition (i)

Let $W_n(\theta) =$ the network delay of the nth cell, and $w_n(\theta) =$ the sample derivative of the nth cell. Note that $G_n(\theta)$ and $g_n(\theta)$ are the empirical averages of $W_n(\theta)$ and $w_n(\theta)$, respectively.

Lemma 5.2

$$\forall n, \lim_{\Delta\theta \longrightarrow 0} \mathbf{E}\left[\frac{\Delta W_n}{\Delta\theta}\right] = \mathbf{E}\left[w_n(\theta)\right]$$

Proof: Let $S^{(n')}(\theta) \in \{\{S\}^{n'}\}$ be the state trajectory for a finite horizon delineated by the n'th network departure, and parameter θ. If $S_1^{(n')}(\theta)$ represents the state trajectory of a nominal path, and $S_1^{(n')}(\theta + \Delta\theta)$ represents the state trajectory of a corresponding perturbed path (having perturbation $\Delta\theta$), then

$$\exists S_2^{(n')}(\theta) s.t. S_2^{(n')}(\theta) = S_1^{(n')}(\theta+\Delta\theta), S_2^{(n')}(\theta+\Delta\theta) = S_1^{(n')}(\theta), \, and \, \mathbf{P}(S_1^{(n')}(\theta)) = \mathbf{P}(S_2^{(n')}(\theta).$$

Therefore $\mathbf{E}\left[W_n(\theta)\right]$ is continuous with respect to θ.

□

Proof of Condition (i):

$$\begin{aligned}
\frac{d}{d\theta}\mathbf{E}\left[G_N(\theta)\right] &= \frac{1}{N}\sum_{n=1}^{N}\frac{d}{d\theta}\mathbf{E}\left[W_n(\theta)\right] \\
&= \frac{1}{N}\sum_{n=1}^{N}\lim_{\Delta\theta\longrightarrow 0}\mathbf{E}\left[\frac{\Delta W_n}{\Delta\theta}\right] \\
&= \frac{1}{N}\sum_{n=1}^{N}\mathbf{E}\left[\lim_{\Delta\theta\longrightarrow 0}\frac{\Delta W_n}{\Delta\theta}\right] \\
&= \frac{1}{N}\sum_{n=1}^{N}\mathbf{E}\left[w_n(\theta)\right] \text{ from Lemma 5.2} \\
&= \mathbf{E}\left[g_n(\theta)\right]
\end{aligned}$$

□

The third equality holds via the Dominated Convergence Theorem, with $\max_i \mathrm{K}_i$ as the bounding function. It is assumed that the network is stable. I.e., the distribution of the state converges to a proper distribution independent of the initial condition. There are only a finite number of jump discontinuities in $\Delta W_n/\Delta\theta$ because there are a limited number of cells that can "affect" a particular cell. I.e., there are a finite number of possible order changes, therefore ΔW_n is constant over an open interval surrounding $\Delta\theta = 0$. This limit of possible order changes is imposed because sojourn times are bounded and input regulation limits the number of cells in the network.

5.3 Condition (ii)

Proof:

$$\begin{aligned}
\lim_{N\longrightarrow\infty} G_N(\theta) &= \lim_{N\longrightarrow\infty}\frac{1}{N}\sum_{n=1}^{N}W_n(\theta), a.s. \\
&= \frac{1}{\lambda}\lim_{t\longrightarrow\infty}\frac{1}{t}\int_0^t (A_s - D_s)\,ds, a.s. \text{ from Little's Theorem} \\
&= \frac{1}{\lambda}\lim_{t\longrightarrow\infty}\frac{1}{t}\int_0^t f(Z_u)\,du, a.s. \\
&= \frac{1}{\lambda}\lim_{t\longrightarrow\infty}\frac{1}{t}\mathbf{E}\left[\int_0^t f(Z_u)\,du\right], a.s. \text{from Prop. 4.3 and } F < \infty \\
&= \frac{1}{\lambda}\lim_{t\longrightarrow\infty}\mathbf{E}(A_t - D_t) \\
&= \lim_{N\longrightarrow\infty}\mathbf{E}\left[W_N(\theta)\right] \text{ from Little's Theorem} \\
&= \lim_{N\longrightarrow\infty}\frac{1}{N}\sum_{n=1}^{N}\mathbf{E}\left[W_n(\theta)\right] \\
&= \lim_{N\longrightarrow\infty}\mathbf{E}\left[G_N(\theta)\right]
\end{aligned}$$

□

5.4 Condition (iii)

Lemma 5.3

$$\mathbf{E}\left[\sum_{n=K_0}^{K_1}\left|\hat{f}\left(Y_n,t_n,\Delta_n\right)\right|\right]<\infty.$$

Proof:

$$\begin{aligned}\mathbf{E}\left[\sum_{n=K_0}^{K_1}\left|\hat{f}\left(Y_n,t_n,\Delta_n\right)\right|\right] &= \mathbf{E}\left[\sum_{n=K_0}^{K_1}\left|f\left(Y_n\right)\left(\tau'_{n+1}-\tau'_n\right)\right|\right]\\ &\leq \mathbf{E}\left[\sum_{n=K_0}^{K_1}F\left|\tau'_{n+1}-\tau'_n\right|\right]\\ &\leq \mathbf{E}\left[\sum_{n=K_0}^{K_1}F^2\right]\\ &\leq F^2\mathbf{E}(h)\\ &< \infty\end{aligned}$$

Note that Proposition 4.5 holds because $\tau'_n(\theta)$ does not go to infinity as n increases. This is due to the cycle times being finite with probability one.

□

Proof of Condition (iii):

$$\begin{aligned}\lim_{n\longrightarrow\infty}g_n\left(\theta\right) &= \lim_{N\longrightarrow\infty}\frac{1}{N}\sum_{n=1}^{N}w_n\left(\theta\right), a.s.\\ &= \lim_{N\longrightarrow\infty}\frac{1}{N}\sum_{n=0}^{N-1}f\left(Y_n\right)\left(\tau'_{n+1}-\tau'_n\right), a.s.\\ &= \lim_{N\longrightarrow\infty}\frac{1}{N}\sum_{n=0}^{N-1}\mathbf{E}\left[f\left(Y_n\right)\left(\tau'_{n+1}-\tau'_n\right)\right], a.s. \text{from Proposition 4.4}\\ &= \lim_{N\longrightarrow\infty}\frac{1}{N}\sum_{n=1}^{N}\mathbf{E}\left[w_n\left(\theta\right)\right]\\ &= \lim_{n\longrightarrow\infty}\mathbf{E}\left[g_n\left(\theta\right)\right]\end{aligned}$$

□

5.5 Condition (iv)

For every realization, let C_N = the number of busy periods after N cells, let h_i = the number of cells in the ith busy period, and let $w_{i,j}(\theta)$ = the gradient estimate for the jth cell in the ith busy period. Keep in mind that the arrival process is Poisson.

Proof:

$$\begin{aligned}\left|\mathbf{E}\left[g_N\left(\theta\right)\right]+b_N\left(\theta\right)\right| &= \left|\mathbf{E}\left[g_N\left(\theta\right)\right]\right|, \text{ where } b_N(\theta)=0\\ &= \left|\frac{1}{N}\sum_{n=1}^{N}\mathbf{E}\left[w_n\left(\theta\right)\right]\right|\\ &= \left|\frac{1}{N}\mathbf{E}\left[\sum_{i=1}^{C_N}\sum_{j=1}^{h_i}w_{i,j}\left(\theta\right)\right]\right|\end{aligned}$$

$$
\begin{aligned}
&\leq \left|\frac{1}{N}\mathbf{E}\left[\sum_{i=1}^{C_N}\sum_{j=1}^{h_i} j\right]\right| \\
&\leq \left|\frac{1}{N}\mathbf{E}\left[\sum_{i=1}^{C_N}\frac{h_i}{2}\left(h_i-1\right)\right]\right| \\
&\leq \frac{1}{2}\left[\mathbf{E}\left(h^2\right)-\mathbf{E}\left(h\right)\right] \\
&< \infty
\end{aligned}
$$

□

6 Conclusions

The strong consistency of an IPA gradient estimate is shown. As a result the efficacy of IPA for gradient estimation is validated.

References

[1] C. A. Brooks and P. Varaiya. Using perturbation analysis to solve the capacity and flow assignment problem for general and ATM networks. In *Proceedings of 3rd International Conference on Telecommunication Systems Modeling and Analysis*, pages 19–32, March 1995.

[2] X.-R. Cao, W.-B. Gong, and Y. Wardi. Ill-conditioned performance functions of queueing systems. *IEEE Transactions on Automatic Control*, 40:1074–1079, 1995.

[3] R. L. Cruz. A calculus for network delay – Parts I and II. *IEEE Transactions on Information Theory*, 37(1):114–141, January 1990.

[4] P. Glasserman. *Gradient Estimation Via Perturbation Analysis.* Kluwer Academic Publishers, 1991.

[5] P. W. Glynn. A GSMP formalism for discrete event systems. *Mathematics of Operations Research*, 5(4):494–501, November 1980.

[6] J.-Q. Hu and S. G. Strickland. Strong consistency of sample path derivative estimates. *Applied Mathematics Letters*, 3(4):55–58, 1990.

[7] L. Kleinrock. *Queueing Systems: Volumes 1 and 2.* Wiley, 1975-1976.

[8] S.-M. Lun, F. Wu, N. Xiao, and P. Varaiya. Netplan: An integrated network planning environment. To appear in "State of the art in Performance Modeling and Simulation, Volume 1: Computer and Communication Networks".

[9] R. Suri. Perturbation analysis: The state of the art and research issues explained via the GI/G/1 queue. *Proceedings of the IEEE*, 77(1):114–137, January 1989.

[10] W. Whitt. Continuity of generalized semi-Markov processes. *Mathematics of Operation Research*, 5(4):494–501, 1980.

[11] N. Xiao, F. F. Wu, and S.-M. Lun. Dynamic bandwidth allocation using perturbation analysis. In *IEEE INFOCOM*, pages 383–389, June 1994.

Applications of Ordinal Optimization to A Robot Motion Planning Problem

Chun-Hung Chen
chchen@eniac.seas.upenn.edu
Department of Systems Engineering

Vijay Kumar
kumar@central.cis.upenn.edu
Department of Mechanical Engineering

University of Pennsylvania
Philadelphia, PA 19104-6315

Abstract

We consider the problem of planning optimal motions for legged robots. Although the motion planning problem has been extensively studied for straight-line locomotion on flat terrain, much less is known about locomotion on uneven terrain with obstacles. In this paper, we discretize the robot motion path and transform the continuous motion problem into a discrete combinatorial optimization problem. Ordinal Optimization is applied to deal with the new problem which is characterized by an extremely large search space. Numerical testing shows that we can quickly isolate a subset of good designs with high probability and reduces the computation time dramatically.

1. Introduction

Ordinal optimization was proposed by Ho, Sreenivas and Vakili (1992). Due to computation budget limitation, we are usually not able to exhaustively search all possible alternatives. Instead of finding the best design, ordinal optimization concentrates on finding a good enough design and can significantly reduce the computation time. This idea has been successfully applied to many stochastic decision problems (Chen 1995), (Ho and Deng 1994), (Ho and Larson 1994). On the other hand, ordinal optimization is equally promising for deterministic combinatorial optimization problems. In this paper, we present an application of ordinal optimization to a robot motion planning problem.

Walking robots have the potential to emulate the superior off-road mobility that biological systems exhibit. For this a walking machine must be able to adapt its gait to the terrain and to select footholds intelligently. This is in contrast to symmetric, periodic and regular gaits that are optimal from the point of view of static stability and have been studied extensively since the early 60's (Tomovic 1961, McGhee 1968). A detailed analysis and classification of the different gaits for quadrupeds and hexapods is found in Song and Waldron (1987). In walking robots, motion planning for walking on unstructured terrain primarily involves (Kumar and Waldron 1989): (a) the determination of the trajectory (direction) of the vehicle; (b) the selection of footholds in space; and (c) the sequencing of the footsteps in time. Chen and Kumar (1996) consider the problem of controlling the gait that includes the sequencing of legs

---

Acknowledgments
This work has been supported by NSF grants BCS92-16691, MSS 91-57156, CISE/CDA 88-22719, ARPA Grant N00014-92-J-1647, and the University of Pennsylvania Research Foundation.

and placing of the feet on appropriate locations on the terrain and formulate the motion planning problem as a discrete optimization problem. Their preliminary simulations show a good motion plan can be isolated quickly with high confidence level.

In this paper we consider a discrete optimization problem formulated by Chen and Kumar (1996). First we show ordinal optimization can efficiently obtain a good enough motion plan in this problem. Secondly, we study the properties of our approaches on the motion planning problem. In the next section we describe the method of ordinal optimization and its advantages. In Section 3, we formulate the motion planning problem for statically-stable machines as a discrete optimization problem. This example is later used in our simulations in Section 4. We show that the ordinal optimization approach adapts the vehicles gait to the terrain and can be used to cross obstacles such as potholes or soft ground that may prevent the leg from stepping on them. When the terrain is free of obstacles, the gait is shown to converge to the wave gait which is well-known to be optimal in terms of stability. In Section 5, we study the properties of our ordinal optimization approaches by considering a reduced motion planning problem.

2. Ordinal Optimization

Given a set $S = \{1, 2, 3, 4, .., |S|\}$, with an associated cost or profit function which we want to optimize. Suppose any design in $G = \{g_1, g_2, g_3, g_4, .., g_{|S|}\}$ is a satisfactory design. Our goal is to find a good design in G quickly. Denote

$p \equiv \frac{|G|}{|S|}$: the population proportion of good designs.

In this paper, we are particularly interested in problems that the total number of designs $|S|$ is extremely large. If we randomly get one sample (by "randomly", we mean that a discrete uniform random number between 1 and $|S|$ is generated to select a design from S), then P(this sample is in G) = p. Suppose we randomly choose m samples without replacement, assuming $|S|$ is much larger than m:

P(at least one of the selected designs is in G)

= 1 - P(none of the selected m designs is in G)

$$= 1 - \left(1 - \frac{|G|}{|S|}\right)\left(1 - \frac{|G|}{|S| - 1}\right) \ldots\ldots \left(1 - \frac{|G|}{|S|-m+1}\right)$$

$$\approx 1 - (1-p)^m \tag{1}$$

If we would like P(at least one of the selected designs is in G) as high as P_{sat}, a satisfactory probability,

$$P_{\text{sat}} = 1 - (1-p)^m \tag{2}$$

Solving (2), we can obtain

$$m = [\ln(1 - P_{\text{sat}})] / \ln(1-p) \tag{3}$$

For example, if we want the probability that the selected best design is within top-0.1% to be as high as 99%, $P_{sat} = 99\%$ and $p = 0.001$. We need only $m = [\ln(1-0.99)] / \ln(1-0.001) = 4602.8 \approx 4603$ samples to achieve such a confidence level for a good design. Note that this number is independent of the size of the sample space $|S|$, provided $|S|$ is much greater than m (e.g., > $10m$). See Table 2.1 for more cases.

Table 2.1 Required number of samples for applying ordinal optimization

P_{sat}	top 1%	top 0.1%	top 0.01%
50%	69	692	6932
80%	160	1608	16094
90%	229	2301	23025
95%	298	2994	29956
98%	389	3910	39119
99%	459	4603	46049
99.9%	687	6904	69075
99.99%	916	9210	92099
99.999%	1146	11508	115124

On the other hand, we can also determine how many samples, on average, we need to have to get a good design. Again, we randomly sample the design space S, and the number of samples is much smaller than $|S|$. Let M be the least number of samples to get a design in G. Thus, M is a random variable with distribution:

$$P(M=m) = P(\text{the first } m\text{-1 samples are not in } G \text{ and the } m\text{-th sample is in } G) = (1-p)^{m-1}p$$

The expectation of M

$$E[M] = \sum_{m=1}^{\infty} m\,(1-p)^{m-1}p = p \sum_{m=1}^{\infty} \frac{d}{dp}[-(1-p)^{m}]$$

$$= p\,\frac{d}{dp}\left[\frac{-(1-p)}{1-(1-p)}\right] = p\,\frac{d}{dp}\left[\frac{p-1}{p}\right] = \frac{1}{p} \qquad (4)$$

For example, if we want a top-0.1% design, $p = 0.001$, $E[M] = \frac{1}{0.001} = 1000$. Therefore, on average, we only need **1000** samples to obtain a **top-0.1%** design.

Note that to obtain the above results, we assume no prior knowledge of the physics of the problem and make no assumption on the structure of the domain while sampling the design space. In robotics applications, we either have some prior knowledge of the design space based on the physics of the problem, or can gradually learn over a period of time. Based on the knowledge of the physics of the problem, a better sampling scheme can be developed. Suppose the probability that a design in G can be obtained using the better sampling scheme is q (we expect that q is much bigger than p). Similarly, from (1),

$$P(\text{at least one of the selected } m \text{ designs is in } G) = 1 - (1-q)^{m} \qquad (5)$$

If we would like P(at least one of the selected designs is in G) as high as P_{sat}, the number of required samples to reach such a satisfactory probability is

$$m = [\ln(1 - P_{sat})] / \ln(1-q) \tag{6}$$

Similar with Eq. (4), the expected least number of samples to get a design in G using the better sampling scheme is

$$E[\bar{M}] = \frac{1}{q} \tag{7}$$

In Section 4, our simulation will show that a "no-knowledge" ordinal optimization based approach yields reasonably good results, while simple heuristics can significantly improve the performance of our approach.

3. A Robot Motion Planning Problem

Consider a planar walking machine with three legs (numbered 1 through 3) moving in a straight line as shown in Figure 3.1. Please see (Chen and Kumar 1996) for details. All legs are assumed to be identical. We discretize the entire workspace. In addition, we consider the intersection of the workspace with the support plane a rectangle and divide the intersection into a finite number of cells. The cell size depends on the size of the foot and the resolution of the terrain map. We assume that the cell size is equal to the foot size.

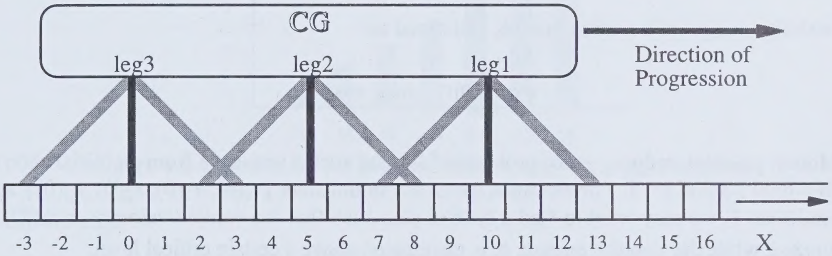


Figure 3.1. A planar model of a walking machine

Define the following coordinates:

$x_c(t)$: the position of the projection of the center of gravity on the support plane at time t.
$x_i(t)$: the position where Leg i is placed at time t, $i = 1, 2, 3$.

We discretize the time interval so that t belongs to the set of non negative integers. The state of the vehicle at time t can be completely described by the set of coordinates $\{x_c(t), x_1(t), x_2(t), x_3(t)\}$. A motion plan for T time intervals is a sequence

$$\{x_c(t), x_1(t), x_2(t), x_3(t)\}, \quad t = 0, 1, ..., T$$

In the example used for simulation we will assume that each leg has a stroke length of 6 units. The pitch is 5 units. Therefore the constraints on the foot placement are given by:

Leg 1: $x_c(t) + 2 \leq x_1(t) \leq x_c(t) + 8$

Leg 2: $x_c(t) - 3 \leq x_2(t) \leq x_c(t) + 3$

Leg 3: $x_c(t) - 8 \leq x_3(t) \leq x_c(t) - 2$

Further, in order to avoid collisions between the legs, we require $x_1(t) > x_2(t) > x_3(t)$. The vehicle velocity constraint limits the velocity of the center of gravity to be less than or equal to a maximum of 3 cell units in each time interval.

At each time instant t, we allow for a leg, i, to be lifted off the ground. (Not more than one leg can be lifted off the ground without compromising the stability of the vehicle). We define a function, $\ell(t)$, which has the value i, if leg i is lifted off the ground, and the value 0, if no leg is lifted off the ground at time t.

In order for the vehicle to be statically stable, the projection of the center of gravity (CG) along the vertical onto the support plane must be within the support polygon. The shortest distance between the projection of the center of gravity (CG) and the support polygon is a measure of static stability and is called the *stability margin* (Song and Waldron 1987). In this planar model, the stability margin between time t and t+1 is

$$sm(t,t+1) = \begin{cases} \min\{x_c(t)-x_2(t),\ x_3(t+1)-x_c(t+1)\}, & \text{if } \ell(t) = 1 \\ \min\{x_c(t)-x_1(t),\ x_3(t+1)-x_c(t+1)\}, & \text{if } \ell(t) = 0 \text{ or } 2 \\ \min\{x_c(t)-x_1(t),\ x_2(t+1)-x_c(t+1)\}, & \text{if } \ell(t) = 3 \end{cases}$$

The stability margin of a motion plan, ϕ, is defined as:

$$\phi \equiv \min_{0 \leq t \leq T-1} sm(t, t+1)$$

Motion planning reduces to the problem of finding such a sequence from an initial state at t=0 to a final state at $t = T$. For example, given the initial state $\{x_c(0), x_1(0), x_2(0), x_3(0)\}$ and the total time T, we may want to find a motion plan such that the vehicle velocity or $x_c(T)$, is maximized, while the stability margin, ϕ, is maintained above a certain critical level.

In this model, the total number of motion plans is extremely large. At time t, we need to decide how many cells the CG will move (4 choices), which leg will be lifted (4 choices), on which foothold to place the leg (7 choices if any leg is lifted). Thus, we have $4*(3*7+1) = 88$ alternatives at any time t. For the complete motion plan, we have 88^T choices. For example, T=25, then 88^{25} is about $4*10^{48}$. Of course, a large number motion plans can be discarded because they do not satisfy the constraints. But the order of feasible motion plans is not much smaller and it is infeasible to search the whole design space to find an optimal motion plan.

4. Simulation

4.1 Strategies for sampling the set of motion plans

There are three basic approaches to sampling the set of motion plans for ordinal optimization, each of which is described below.

A. Uniform Approach.

If we do not possess any knowledge of the physics of the problem or the constraints, we may adopt the uniform approach in which we sample the design space or the set of motion plans using a uniform distribution. In other words, the probability of sampling a particular candidate plan for evaluation is the same as that of selecting any other plan. For example, when selecting which leg to lift at any time *t*, the probabilities of lifting Leg 1, Leg 2, or Leg 3, or of not lifting any leg are all the same (=0.25). During foothold selection, when we have to select one of 7 different footholds, say, Cells 10, 11, .., 16, to place a lifted leg, we generate a random number using a discrete uniform distribution: $p(i=10) = p(i=11) = p(i=12) = .. = p(i=16) = 1/7$.

B. Heuristic Approach.

We can use our intuition or our knowledge of the physics of the problem to bias our sampling of the set of motion plans. For example, in the example of foothold selection, since we want to maximize the moving distance, we may discriminate against footholds that are not towards the front of the vehicle (in the direction of progression). At the same time, we do not want to exclude other foot-holds. Thus, we arrive at a simple heuristic. When sampling the set of motion plans, the sampling of foot-holds (between Cells 10-16 in the example) is determined by using the distribution shown in Figure 4.1.

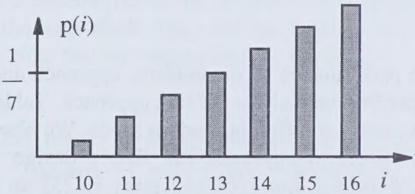


Figure 4.1. Distribution for heuristic approach to sampling footholds.

C. Off-line Approach

Finally, we may need to evaluate the uniform or the heuristic approach by comparing their performances with the optimal design. However, even though the optimal motion plan for evaluation can be found off-line, the large search space (the number of motion plans, $|S|$, $\approx 4 * 10^{48}$) precludes an exhaustive search. From Table 2.1, we know that even the uniform approach can obtain a top-0.01% plan with 99.999% confidence level after 115,000 samples. We can expect the heuristic approach to have better performance (we will show this is true latter). Thus, we can apply the heuristic approach with a large number (greater than a billion) samples to arrive at an *almost-optimal* plan which is very close to the optimal motion plan (certainly within less than 0.01%) with a very high confidence level (better than 99.999%). Clearly this approach is only feasible for off-line planning or evaluation.

4.2 Straight line locomotion on smooth terrain

This is a well studied problem and analytical solutions for optimal gaits can be found in the literature (Song and Waldron 1987, Kumar 1987). There are no obstacles on the terrain. We consider T=25 in this testing. Our objective is to maximize $x_c(25)$ subject to $\phi \geq 1$. We use the off-line approach to find a design which is very close to the optimal design. The result which yields $x_c^*(25) = 42$ is shown in Figure 4.2.

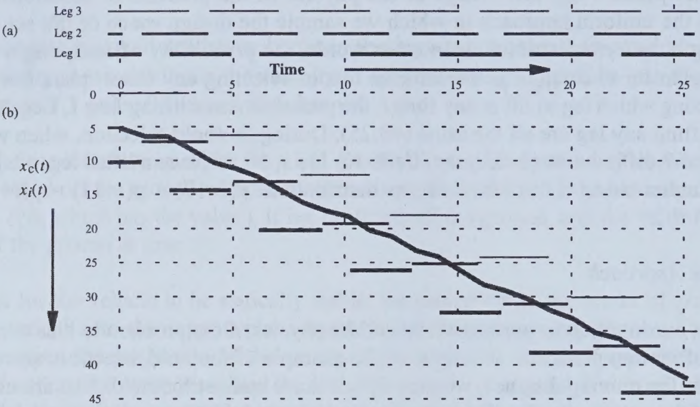


Figure 4.2. Almost-optimal motion plan for smooth-terrain locomotion. In the gait diagram for the plan, shown in 4.2(a), a solid line (the three different types denote the different legs) shows the time instants for which a leg is in support phase. In 4.2 (b), the coordinates of the center of gravity, $x_c(t)$, denoted by the gray line, and those of the support legs, $x_i(t)$, are plotted against time.

We now compare the performance of the uniform approach and the heuristic approach with the almost-optimal performance of the off-line approach. Table 4.1 contains the results after testing 3000 motion plans for different random seeds. We observe that with only 3000 samples, on average, the uniform approach can find a design in which $x_c(25)=21.8$. Furthermore, the simple heuristic approach can push $x_c(25)$ up to 36.6 with only 3000 samples. The computation time on a SUN-4 workstation is only 2 seconds. Given that the total number of candidate motion plans is $|S| = 4 * 10^{48}$, this is a tremendous savings in terms of computational load.

Table 4.1. Straight line locomotion on smooth terrain.

Random Seed	max $x_c(25)$ (Uniform)	max $x_c(25)$ (Heuristic)
1	25	37
12	21	35
123	20	37
1234	21	36
12345	23	38
123456	21	37
AVERAGE	**21.8**	**36.6**

4.3 Crossing a known large obstacle

Given the physical limitations of the three-legged planar robot that we have chosen (stroke length is less than or equal to six units), the largest ditch the robot can cross is five units wide. We explore the performance of our methods in the presence of such a ditch which occupies Cells 15 - 19. The objective is to maximize the distance traveled in 25 time units. The almost-optimal plan obtained using the off-line approach is shown in Figure 4.3 in which $x_c^*(25)=34$. Table 4.2 contains the results of the uniform and heuristic approaches applied to a sample of 30000 motion plans. The heuristic approach yields a result that is very close and sometimes, the same as the plan found off-line.

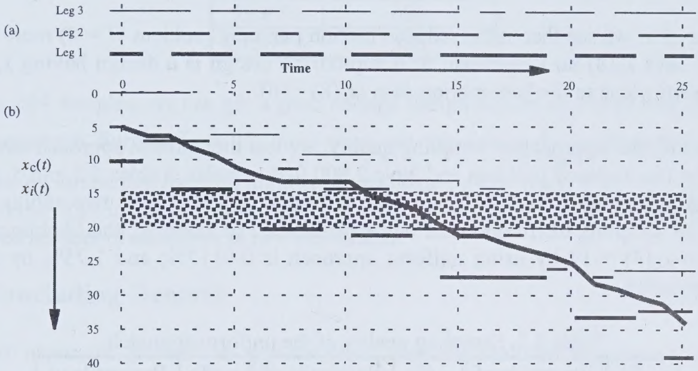


Figure 4.3. Almost-optimal motion plan to the large-obstacle problem. The horizontal shaded area $19 \geq x \geq 15$ denotes the large ditch. The solid black lines and the gray line has the same meaning as in Figure 4.2. Note that no foot is placed in the ditch.

Table 4.2. Simulation results for straight line locomotion crossing a large ditch.

Random Seed	max $x_c(25)$ (Uniform)	max $x_c(25)$ (Heuristic)
1	19	34
12	17	33
123	19	31
1234	19	34
12345	18	31
123456	19	33
AVERAGE	**18.5**	**32.7**

5. Property Studies of Ordinal Optimization on the Robot Motion Planning Problem

In the motion problem $T = 25$, the large space size $|S| \approx 4 * 10^{48}$ precludes an exhaust search. To study the properties of our ordinal approaches on the motion planning problems, we consider a much smaller problem in which $T = 8$. By studying the smaller problem, we can have better ideas on the properties of the motion planning problem. In this reduced problem, $|S| \approx 3 * 10^{15}$. The computation time to exhaustively search the whole space is about three days. Table 5.2 shows the population percentiles for different $x_c(8)$'s.

Table 5.1. The population percentiles of designs for different $x_c(8)$'s.

$x_c(8)$	Percentile of designs	$x_c(8)$	Percentile of designs	$x_c(8)$	Percentile of designs
5	1.06%	9	12.6%	13	0.388%
6	6.91%	10	9.14%	14	0.0419%
7	21.2%	11	4.66%	15	0.00036%
8	42.3%	12	1.70%	16	0.000037%

Form Table, 5.1, we see that in the reduced motion planning problem ($T = 8$) more than 70% of designs have $x_c(8)$ no larger than 8. A top-0.01% design is a design having $x_c(8) \geq 14$, which is pretty close to the best motion plan ($x_c(8) = 16$).

To quantify our approaches' sampling quality, we test the uniform approach and heuristic approach on this reduced problem and have 2,000,000 samples. Tables 5.2 and 5.3 contains the percentiles of the sampled designs for different $x_c(8)$'s. These two tables can well represent the sampling distributions of our approaches. For example, the likelihood to get a design with $x_c(8) = 14$ by using uniform approach is 0.0117%; and 1.75% by using our heuristic approach.

Table 5.2. Sampling quality of the uniform approach.

$x_c(8)$	Percentile of designs	$x_c(8)$	Percentile of designs	$x_c(8)$	Percentile of designs
5	0.795%	9	5.55%	13	0.10%
6	5.74%	10	3.99%	14	0.0117%
7	22.1%	11	1.93%	15	0.00008%
8	59.0%	12	0.693%	16	0.000005%

Table 5.3. Sampling quality of the heuristic approach.

$x_c(8)$	Percentile of designs	$x_c(8)$	Percentile of designs	$x_c(8)$	Percentile of designs
5	0.386%	9	11.9%	13	3.64%
6	2.87%	10	14.3%	14	1.75%
7	11.6%	11	12.1%	15	0.45%
8	32.5%	12	8.34%	16	0.06%

Our uniform approach puts equal chance for all feasible selections at each time instant. However, this doesn't imply the uniformity of design selection in the whole design space. More specifically, if we are able to label a number to each design and then generate a discrete uniform random number between 1 and $|S|$ to select a design from S, the sampling distribution will be the same as Table 5.1. There are some differences between Tables 5.1 and 5.2, although the differences are not significant.

On the other hand, Table 5.3 shows that even the simple heuristic approach can significantly increase the probability to get a good design. For example, if any design with $x_c(8) \geq 14$ is considered to be a good design, the heuristic approach has 2.26% chance to get a good design for each sample, while only 0.042% of the total population have $x_c(8) \geq 14$. The derivations in Section 2 yields Table 5.4 showing the required number of samples to obtain a design $x_c(8) \geq 14$ with confidence level P_{sat} using the simple heuristic approach.

Table 5.4. Required number of samples to apply the heuristic approach to obtain a design $x_c(8) \geq 14$ with confidence level P_{sat}.

P_{sat}	# of samples
50%	31
80%	71
90%	101
99%	202
99.99%	403
99.999%	504

With only 504 samples, we can get a good enough design almost sure even using the simple heuristic approach. Further, from (7), on average, we only need $\frac{1}{0.026} \approx$ **39** samples to obtain a good design. Given that the total number of candidate motion plans is $|S| = 3 * 10^{15}$, this is a tremendous savings in terms of computational load. If we develop a better sampling scheme, the required number of samples can be even smaller.

6. Concluding Remark

In this paper we present an application of Ordinal Optimization to a robot motion planning problem. Our simulation shows that a "no-knowledge" ordinal optimization based approach yields reasonably good results, while simple heuristics can significantly improve the performance of our approach. We also apply a simplified problem to quantify our sampling techniques. We firmly believe that there exist better sampling schemes. The development of a better sampling scheme and the quantification of its sampling quality is a subject for ongoing research.

References

[1] Chen, C. H., "An Effective Approach to Smartly Allocate Computing Budget for Discrete Event Simulation," To appear in the *Proceedings of the 34th Conference of Decision and Control*, 1995.

[2] Chen, C. H., and Kumar, V. "Motion Planning of Walking Robots in Environments with Uncertainty," submitted to *1996 IEEE International Conference on Robotics and Automation* (Also available as: Working Paper #95-12, Department of Systems Engineering, University of Pennsylvania).

[3] Ho, Y. C., and Deng, M., "The Problem of Large Search Space in Stochastic Optimization," *Proceedings of the 33rd Conference of Decision and Control*, 1994.

[4] Ho, Y. C., Sreenivas, R. S., and Vakili, P., "Ordinal Optimization of DEDS", *Journal of Discrete Event Dynamic Systems*, 2, #2, pp. 61-88, 1992.

[5] Kumar, V. 1987. Motion Planning in Legged Locomotion Systems. *Ph.D. Dissertation*. Department of Mechanical Engineering, The Ohio State University.

[6] Kumar, V., Waldron, K.J., 1989. Adaptive Gait Control for a Walking Robot. *Journal of Robotic Systems*. Vol. 6, No. 1. pp. 49-75.

[7] Kumar, V. and Waldron, K.J., "Force Distribution in Walking Vehicles on Uneven Terrain," *ASME Journal of Mechanisms, Transmissions and Automation in Design*, Vol. 112, No. 1, 1990, pp. 90-99.

[8] Song, S.M., and Waldron, K.J., 1987. An analytical approach for gait study and its application on wave gaits. *Int. J. Robotics Research*. Vol. 6. No. 2.

Fault Detection and Identification in Communication Networks: a Discrete Event Systems Approach

Yongseok Park and Edwin K. P. Chong
School of Electrical and Computer Engineering
Purdue University
West Lafayette, Indiana 47907-1285
e-mail: yongseok, echong@ecn.purdue.edu

Abstract

We consider the problem of fault detection and identification in communication networks, where communication protocols are modeled as finite state machines (FSMs). Faults are specified as change and addition of arcs in the FSM. We give necessary and sufficient conditions for a fault to be detectable and identifiable with a given delay. We also provide an efficient algorithm for performing on-line detection and identification of faults.

1 Introduction

Communication networks have experienced a dramatic increase in size and complexity in recent years. A typical communication network consists of thousands of hardware devices, such as switches, routers, adapters, and workstations. Failure of devices in a communication network causes not only inconvenience, but may also be hazardous. Therefore, there has been significant interest in methods for fast fault detection and recovery in communication networks.

There have been many approaches to the problems of fault detection, identification, and diagnosis. Recently, several model-based approaches using discrete event systems have been reported: in communication network protocols [3], [4], [2], manufacturing systems [5], [1], heating, ventilation, and air conditioning systems [16], exhaust gas recirculation systems [7], and nuclear plants [13].

In [3], [4], Bouloutas *et al.* consider the problem of fault detection in communication processes modeled as finite state machines (FSMs). They show that given a system FSM, a faulty system FSM, and an observer FSM, whether the observer can detect the fault or not can be tested in polynomial complexity (in the number of states and events). In [15], Sampath *et al.* investigate the problem of fault diagnosability in discrete event systems, which, roughly speaking, is the property that unobservable faults can be detected and identified with bounded delay based on future observations of the system.

In this paper, we consider the two problems of fault detection and identification with a given delay bound. We use a discrete event system model based on [8], where

---

This work was supported in part by a grant from the Purdue NSF Engineering Research Center for Collaborative Manufacturing and by the National Science Foundation through grants ECS-9410313 and ECS-9501652.

communication protocols are modeled as FSMs. Faults are specified as change and addition of arcs in the FSM, as in [2]. We give structural conditions for a given fault to be detectable and identifiable. We characterize the maximum delay in detection and identification of faults, which is important in real-time on-line monitoring [17], and can be a factor in design of monitoring systems. We also provide an efficient algorithm for on-line fault detection and identification. Our method is applicable to any system modeled as an FSM where faults are specified as change and addition of arcs.

2 Formal Model

2.1 FSM model

We follow the FSM model of [8] for communication protocols. The model is represented by the 6-tuple $\mathcal{G} = \{S, I, O, \delta_{\mathcal{G}}, \rho_{\mathcal{G}}, s(0)\}$, where

- S is a finite set of states,
- I is a finite set of inputs,
- O is a finite set of outputs,
- $\delta_{\mathcal{G}} : S \times I \to S$ is the partial state transition function,
- $\rho_{\mathcal{G}} : S \times I \to O$ is the partial output function,
- $s(0)$ is an initial state.

We assume that the FSM is deterministic in the sense that for each state and input, there is at most one next state, specified by $\delta_{\mathcal{G}}$. The transition function $\delta_{\mathcal{G}}$ is a partial function in the sense that for each s, $\delta_{\mathcal{G}}(s, \cdot)$ is defined on a subset $I_{\mathcal{G}}(s)$ of I. Elements of $I_{\mathcal{G}}(s)$ are called the feasible inputs at state s. The above model is also called a Mealy machine [6].

The FSM is visualized by a directed graph $G = (V, E)$, where $V = S$ is the set of vertices representing states, and E is the set of arcs (v_i, v_j) with an input/output label i/o, which represents a transition from state v_i to v_j with the input i and the output o. We refer to arcs in the graph G as *events*. Given an event $e \in E$, let e_{ori}, e_{in}, e_{out}, e_{des} represent the *origin state*, *input*, *output*, and *destination state*, respectively, of event e.

2.2 Fault model

We consider the following five types of faults, specified as change and addition of arcs in the system graph G.

1. The output of an arc is altered.
2. The input of an arc is altered (without causing nondeterminism).
3. The destination of an arc is altered.
4. The origin of an arc is altered (without causing nondeterminism).
5. An arc is added (without causing nondeterminism).

The first four types of faults are considered in [8] for conformance testing of communication protocols, while the last type of fault is considered in [15].

Let E_F denote the set of specified faulty events. Each faulty event corresponds to an arc not in the system graph G, i.e., $E_F \cap E = \emptyset$. Let $F = \cup_{i=1}^{5} F_i$ denote the set of specified faults, where F_i is a prespecified set of faults of type i. Each fault in F is represented by an ordered pair (e, f) where $e \in E \cup \{\epsilon\}$ and $f \in E_F$. The set F_i are described as follows:

1. $F_1 = \{(e, f) \in E \times E_F : e_{out} \neq f_{out}, e_j = f_j, j \neq \text{ out}\}$,
2. $F_2 = \{(e, f) \in E \times E_F : e_{in} \neq f_{in}, e_j = f_j, j \neq \text{ in}, f_{in} \neq I_{\mathcal{G}}(f_{ori})\}$,
3. $F_3 = \{(e, f) \in E \times E_F : e_{des} \neq f_{des}, e_j \neq f_j, j \neq \text{ des}\}$,
4. $F_4 = \{(e, f) \in E \times E_F : e_{ori} \neq f_{ori}, e_j = f_j, j \neq \text{ ori}, f_{in} \notin I_{\mathcal{G}}(f_{ori})\}$,
5. $F_5 = \{(\epsilon, f) : f \in E_F : f_{in} \notin I_{\mathcal{G}}(f_{ori})\}$.

For each fault $(e, f) \in F$, we construct a faulty FSM $\mathcal{G}_f = \{S, I, O, \delta_{\mathcal{G}_f}, \rho_{\mathcal{G}_f}, s(0)\}$, where $\delta_{\mathcal{G}_f}$ and $\rho_{\mathcal{G}_f}$ are the accordingly modified version of $\delta_{\mathcal{G}}$ and $\rho_{\mathcal{G}}$, respectively. The FSM $\mathcal{G}$ changes to $\mathcal{G}_f$ when the faulty event f occurs at state f_{ori}.

3 Fault Detection and Identification

We first define the following notation. Given a FSM $\mathcal{A}$ with state space S, and two states $s_1, s_2 \in S$, let $L(\mathcal{A}, s_1, s_2)$ denote the set of all possible input/output traces in system $\mathcal{A}$ starting from state s_1 and terminating at state s_2, given by

$$\begin{aligned} L(\mathcal{A}, s_1, s_2) &= \{(i(1)/o(1)) \cdots (i(k)/o(k)) \in (E \times O)^* : k \in \mathbb{N}, i(1) \in I_{\mathcal{A}}(s_1), \\ &\quad s(1) = s_1, \delta_{\mathcal{A}}(s(j), i(j)) = s(j+1), o(j) = \rho_{\mathcal{A}}(s(j), i(j)), j \in [1, k], \\ &\quad s(k+1)) = s_2\} \cup \{\epsilon\}. \end{aligned}$$

Similarly, let $L(\mathcal{A}, s, *)$ and $L(\mathcal{A}, *, s)$ denote the set of all traces starting from and terminating at state s, respectively, given by

$$L(\mathcal{A}, s, *) = \{\sigma \in L(\mathcal{A}, s, s') : s' \in S\},$$

$$L(\mathcal{A}, *, s) = \{\sigma \in L(\mathcal{A}, s', s) : s' \in S\}.$$

In our framework, no states are *directly* observable, but input/output pairs are observed through an observation map. Let $\Lambda_{in} : I \rightarrow \Gamma_{in} \cup \{\epsilon\}$ and $\Lambda_{out} : O \rightarrow \Gamma_{out} \cup \{\epsilon\}$ denote the input observation map and the output observation map, respectively, where Γ_{in} is the set of input observation symbols, Γ_{out} the set of output observation symbols, and ϵ is the *null observation.* The overall observation map Λ is then given by

$$\Lambda(i/o) = \begin{cases} (\Lambda_{in}(i), \Lambda_{out}(o)) & \text{if } \Lambda_{in}(i) \neq \epsilon \text{ or } \Lambda_{out}(o) \neq \epsilon \\ \epsilon & \text{otherwise.} \end{cases}$$

We extend Λ to input/output traces as follows:

$$\Lambda((i_1/o_1) \cdots (i_k/o_k)) = \Lambda(i_1/o_1) \cdots \Lambda(i_k/o_k).$$

Similarly, we can extend Λ to sets of input/output traces in the obvious way. The null observation ϵ satisfies $\epsilon \cdot \mu = \mu \cdot \epsilon = \mu$.

Consider the system $\mathcal{G}$ and a faulty event $f \in E_F$. Define

$$L_f(d) = \{\mu \cdot (f_{in}/f_{out}) \cdot \nu : \mu \in L(\mathcal{G}, s(0), f_{ori}), \nu \in L(\mathcal{G}_f, f_{des}, *), |\nu| \geq d\}.$$

Each trace in $L_f(d)$ is a concatenation of a trace from $L(\mathcal{G}, s(0), f_{ori})$, the pair (f_{in}/f_{out}), and a trace from $L(\mathcal{G}_f, f_{des}, *)$ with length d or greater. A trace in $L_f(d)$ represents input/output pairs that can occur in a system where the faulty event f occurs.

Definition 1 A fault $(e, f) \in F$ is said to be *d-detectable* if

$$\Lambda(L(\mathcal{G}, s(0), *)) \cap \Lambda(L_f(d)) = \emptyset.$$

A fault is d-detectable if we can distinguish a faulty system from the nominal system via observations of the input/output traces up to at most d events after the occurrence of the fault.

Definition 2 A fault $(e, f) \in F$ is said to be *d-identifiable* if

1. (e, f) is d-detectable, and
2. for each $f' \in F \setminus \{f\}$, we have $\Lambda(L_{f'}(0)) \cap \Lambda(L_f(d)) = \emptyset$.

A fault is d-identifiable if, in addition to being d-detectable, the faulty system corresponding to the given fault can be distinguished from all other possible faulty systems via observations of the input/output traces up to at most d events after the occurrence of the fault.

3.1 State distinguishability

To investigate detectability and identifiability of faults, we need the concept of distinguishability between states, which is considered in [14], [10], [9], [12], [11]. In this subsection, we introduce the notion of distinguishability between states in two different FSMs.

Let $\mathcal{A}$, $\mathcal{B}$ be two FSMs with the same state space S.

Definition 3 A state pair $(s, s') \in S \times S$ is said to be *d-distinguishable* in $(\mathcal{A}, \mathcal{B})$ if for each pair $(\mu, \nu) \in L(\mathcal{A}, s, *) \times L(\mathcal{B}, s', *)$, we have $|\nu| \geq d \Rightarrow \Lambda(\mu) \neq \Lambda(\nu)$.

A state pair (s, s') is d-distinguishable in $(\mathcal{A}, \mathcal{B})$ if the observation of every input/output trace starting at state s in $\mathcal{A}$ with length greater than equal to d, is different from the observation of every input/output trace starting at state s' in $\mathcal{B}$. The property of d-distinguishability between states in the same system is considered in [11]. Note that d-distinguishability is not symmetric, that is, (s_1, s_2) being d-distinguishable does not necessarily imply that (s_2, s_1) is d-distinguishable.

We need to define the following notation to proceed. Let $D_F(\mathcal{A}, \mathcal{B}, d)$ denote the set of state pairs that are d-distinguishable in $(\mathcal{A}, \mathcal{B})$. Let $Q(\mathcal{A}, \mathcal{B}, s_1, s_2)$ denote the set of state pairs that are reachable from s_1 in $\mathcal{A}$ and s_2 in $\mathcal{B}$, respectively, by events with the same observation, given by

$$\begin{aligned} Q(\mathcal{A}, \mathcal{B}, s_1, s_2) &= \{\delta_{\mathcal{A}}(s_1, i_1), \delta_{\mathcal{B}}(s_2, i_2) : i_1 \in I_{\mathcal{A}}(s_1), i_2 \in I_{\mathcal{B}}(s_2), \\ &\quad \Lambda(i_1/\rho_{\mathcal{A}}(s_1, i_1)) = \Lambda(i_2/\rho_{\mathcal{B}}(s_2, i_2))\}. \end{aligned}$$

Let $Q_\epsilon(\mathcal{A}, s)$ denote the set of states that are reachable from s in $\mathcal{A}$ by an event with the null observation, given by

$$Q_\epsilon(\mathcal{A}, s) = \{\delta_A(s, i) : i \in I_A(s), \Lambda(i/\rho_\mathcal{A}(s, i)) = \epsilon\}.$$

The following proposition relates d-distinguishable state pairs and their neighboring state pairs.

Proposition 1 A state pair $(s_1, s_2) \in S \times S$ is d-distinguishable in $(\mathcal{A}, \mathcal{B})$ if and only if the following three conditions hold:

1. $Q(\mathcal{A}, \mathcal{B}, s_1, s_2) \subset D_F(\mathcal{A}, \mathcal{B}, d-1)$;
2. $Q_\epsilon(\mathcal{A}, s_1) \times \{s_2\} \subset D_F(\mathcal{A}, \mathcal{B}, d-1)$;
3. $\{s_1\} \times Q_\epsilon(\mathcal{B}, s_2) \subset D_F(\mathcal{A}, \mathcal{B}, d)$.

Proof: Sufficiency: Pick arbitrary input/output traces $\mu \in L(\mathcal{A}, s_1, *)$ and $\nu \in L(\mathcal{B}, s_2, *)$ such that $|\mu| \geq d$. Let μ be written as $\mu = (i_1/\rho_\mathcal{A}(s_1, i_1)) \cdot \mu'$. If $\Lambda(i_1/\rho_\mathcal{A}(s_1, i_1)) = \epsilon$, then, by Condition 2, we have $\Lambda(\mu) = \Lambda(\mu') \neq \Lambda(\nu)$. Suppose that $\Lambda(i_1/\rho_\mathcal{A}(s_1, i_1)) \neq \epsilon$. If $|\nu| = 0$, then it is evident that $\Lambda(\mu) \neq \Lambda(\nu)$. Let ν be written as $\nu = (i_2/\rho_\mathcal{B}(s_2, i_2)) \cdot \nu'$. There are three cases to consider. First, suppose that $\Lambda(i_1/\rho_\mathcal{A}(s_1, i_1)) = \Lambda(i_2/\rho_\mathcal{B}(s_2, i_2))$. Then, by Condition 1, we have $\Lambda(\mu') \neq \Lambda(\nu')$. Therefore, we have $\Lambda(\mu) \neq \Lambda(\nu)$. Second, suppose that $\Lambda(i_1/\rho_\mathcal{A}(s_1, i_1)) \neq \Lambda(i_2/\rho_\mathcal{B}(s_2, i_2))$, and $\Lambda(i_2/\rho_\mathcal{B}(s_2, i_2)) \neq \epsilon$. Then, obviously we have $\Lambda(\mu) \neq \Lambda(\nu)$. Finally, suppose that $\Lambda(i_1/\rho_\mathcal{A}(s_1, i_1)) \neq \Lambda(i_2/\rho_\mathcal{B}(s_2, i_2))$, and $\Lambda(i_2/\rho_\mathcal{B}(s_2, i_2)) = \epsilon$. Then, by Condition 3, we have $\Lambda(\mu) \neq \Lambda(\nu') = \Lambda(\nu)$.

Necessity: Pick arbitrary input/output traces $\mu \in L(\mathcal{A}, s_1, *)$ and $\nu \in L(\mathcal{B}, s_2, *)$ such that $|\mu| \geq d$ and $|\nu| \geq 1$. Suppose that we have $\Lambda(\mu) \neq \Lambda(\nu)$. Let μ and ν be written as $\mu = (i_1/\rho_\mathcal{A}(s_1, i_1)) \cdot \mu'$ and $\nu = (i_2/\rho_\mathcal{B}(s_2, i_2)) \cdot \nu'$. First, if $\Lambda(i_1/\rho_\mathcal{A}(s_1, i_1)) = \Lambda(i_2/\rho_\mathcal{B}(s_2, i_2))$, then we have $\Lambda(\mu') \neq \Lambda(\nu')$. Since $|\mu'| \geq d-1$, Condition 1 holds. Second, if $\Lambda(i_1/\rho_\mathcal{A}(s_1, i_1)) = \epsilon$, then we have $\Lambda(\mu') \neq \Lambda(\nu)$. Therefore, Condition 2 holds. Third, if $\Lambda(i_2/\rho_\mathcal{B}(s_2, i_2)) = \epsilon$, then we have $\Lambda(\mu) \neq \Lambda(\nu')$. Therefore, Condition 3 holds. ∎

The following algorithm computes the set $D_F(\mathcal{A}, \mathcal{B}, d)$, based on Proposition 1.

Algorithm 1 *d-Distinguishable state pairs in* $(\mathcal{A}, \mathcal{B})$

```
$U \leftarrow \{(s_1, s_2) : (s_1, s_2) \in S \times S\}$
$D_F(\mathcal{A}, \mathcal{B}, 0) \leftarrow \emptyset$
for $i = 1$ to $d$
    $temp(0) \leftarrow D_F(\mathcal{A}, \mathcal{B}, i-1)$
    $temp(1) \leftarrow temp(0)$
    for each $(s_1, s_2) \in U \setminus temp(0)$
        if $Q(\mathcal{A}, \mathcal{B}, s_1, s_2) \subset D_F(\mathcal{A}, \mathcal{B}, i-1)$ and $Q_\epsilon(\mathcal{A}, s_1) \times \{s_2\} \subset D_F(\mathcal{A}, \mathcal{B}, i-1)$
        and $\{s_1\} \times Q_\epsilon(\mathcal{B}, s_2) \subset temp(1)$
            then $temp(1) \leftarrow temp(1) \cup \{(s_1, s_2)\}$
    $j \leftarrow 1$
    while $temp(j) \neq temp(j-1)$
        $j \leftarrow j + 1$
        for each $(s_1, s_2) \in U \setminus temp(j-1)$
        if $Q(\mathcal{A},\mathcal{B},s_1,s_2) \subset D_F(\mathcal{A},\mathcal{B},d-1)$ and $Q_\epsilon(\mathcal{A},s_1)\times\{s_2\} \subset D_F(\mathcal{A},\mathcal{B},d-1)$ and $\{s_1\}\times Q_\epsilon(\mathcal{B},s_2) \subset temp(j)$
            then $temp(j) \leftarrow temp(j) \cup \{(s_1,s_2)\}$
    $D_F(\mathcal{A},\mathcal{B},i) \leftarrow temp(j)$
  return $D_F(\mathcal{A},\mathcal{B},d)$
end
```

While d-distinguishability concerns "future" observation traces, we can similarly define a notion of distinguishability that uses "past" observation traces to distinguish between states. We refer to this notion as backward-distinguishability, which is considered in [12].

Definition 4 A state pair $(s_1, s_2) \in S \times S$ is said to be *backward distinguishable* in system $\mathcal{A}$ if for each pair $(\mu,\nu) \in L(\mathcal{A}, s(0), s_1) \times L(\mathcal{A}, s(0), s_2)$, we have $\Lambda(\mu) \neq \Lambda(\nu)$.

An algorithm to compute the set of backward distinguishable state pairs can be found in [12].

3.2 Fault d-detectability

In this subsection, we give a necessary and sufficient condition for a fault to be d-detectable. Let $D_B(\mathcal{A})$ denote the set of backward distinguishable state pairs in $\mathcal{A}$.

Theorem 1 A fault $(e, f) \in F$ is d-detectable if and only if for each $s \in S$ such that $(s, f_{ori}) \notin D_B(\mathcal{G})$ and each $i \in I_\mathcal{G}(s)$, we have $\Lambda(i/\rho_\mathcal{G}(s)) = \Lambda(f_{in}/f_{out}) \Rightarrow (f_{des}, \delta_\mathcal{G}(s,i)) \in D_F(\mathcal{G}_f, \mathcal{G}, d)$.

Proof: Sufficiency: There are two cases to consider. First, consider $s \in S$ such that $(s, f_{ori}) \notin D_B(\mathcal{G})$ and there exists an input i such that $i \in I_\mathcal{G}(s)$ and $\Lambda(i/\rho_\mathcal{G}(s,i)) = \Lambda(f_{in}/f_{out})$. Pick $\omega = \mu \cdot (f_{in}/f_{out}) \cdot \nu \in L_f(d)$ and $\omega' = \mu' \cdot (i/\rho_\mathcal{G}(s,i)) \cdot \nu' \in L(\mathcal{G}, s(0), *)$ such that $\mu \in L(\mathcal{G}, s(0), f_{ori})$, $\nu \in L(\mathcal{G}_f, f_{des}, *)$, $|\nu| \geq d$, $\mu' \in L(\mathcal{G}, s(0), s)$, and $\nu' \in L(\mathcal{G}, \delta_\mathcal{G}(s,i), *)$. Then, it is evident that $\Lambda(\omega) \neq \Lambda(\omega')$, since we have $\Lambda(\mu) = \Lambda(\mu')$, $\Lambda(i/\rho_\mathcal{G}(s,i)) = \Lambda(f_{in}/f_{out})$, and $\Lambda(\nu) \neq \Lambda(\nu')$ by assumption. Second, consider $s \in S$ such that $(s, f_{ori}) \notin D_B(\mathcal{G})$ but there does not exist an input i such that $i \in I_\mathcal{G}(s)$ and $\Lambda(i/\rho_\mathcal{G}(s,i)) = \Lambda(f_{in}/f_{out})$. Then, for each $\omega = \mu \cdot (f_{in}/f_{out}) \cdot \nu \in L_f(d)$, there does not exist $\omega' = \mu' \cdot (i'/o') \cdot \nu' \in L(\mathcal{G}, s(0), *)$ such that $\mu' \in L(\mathcal{G}, s(0), s)$, $\nu' \in L(\mathcal{G}, s, *)$, and $\Lambda(\omega') = \Lambda(\omega)$, since we have $\Lambda(\mu) = \Lambda(\mu')$ and $\Lambda(f_{in}/f_{out}) \neq \Lambda(i'/o')$.

Necessity: Suppose fault $(e, f) \in F$ is d-detectable, that is, $\Lambda(L(\mathcal{G}, s(0), *)) \cap \Lambda(L_f(d)) = \emptyset$. Pick an arbitrary state $s \in S$ and $i \in I_\mathcal{G}(s)$ such that $(s, f_{ori}) \notin D_B(\mathcal{G})$ and $\Lambda(i/\rho_\mathcal{G}(s,i)) = \Lambda(f_{in}/f_{out})$. By assumption, there exist $\mu \in L(\mathcal{G}, s(0), s)$ and $\mu' \in L(\mathcal{G}, s(0), f_{ori})$ such that $\Lambda(\mu) = \Lambda(\mu')$. Let $\omega = \mu \cdot (i/\rho_\mathcal{G}(s,i)) \cdot \nu$ and $\omega' = \mu' \cdot (f_{in}/f_{out}) \cdot \nu'$ such that $\nu \in L(\mathcal{G}, \delta_\mathcal{G}(s,i), *)$, $\nu' \in L(\mathcal{G}_f, f_{des}, *)$, and $|\nu'| \geq d$. By assumption, we have $\Lambda(\omega) \neq \Lambda(\omega')$, $\Lambda(\mu) = \Lambda(\mu')$, and $\Lambda(i/\rho_\mathcal{G}(s,i)) = \Lambda(f_{in}/f_{out})$. Therefore, we have $\Lambda(\nu) \neq \Lambda(\nu')$, and hence $(f_{des}, \delta_\mathcal{G}(s,i)) \in D_F(\mathcal{G}_f, \mathcal{G}, d)$. ∎

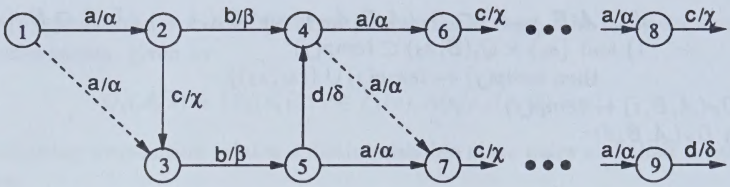


Figure 1: A case where faults are not identifiable

3.3 Fault d-identifiability

The conditions in Theorem 1 only guarantee that if a fault has occurred, then we can deduce that *some* fault occurred, but not necessarily the identity of the fault. It is possible that two faults may lead to the same observation trace. Figure 1, for example, illustrates this possibility with faults $((1, a/\alpha, 2), (1, a/\alpha, 3))$ and $((4, a/\alpha, 6), (4, a/\alpha, 7))$ (where the notation $(s_1, i/o, s_2)$ represents an arc from state s_1 to s_2 with input i and output o).

The following theorem characterizes a necessary and sufficient condition for a fault, if it occurs, to be identifiable via observations of at most d subsequent events. Recall that a fault is d-identifiable if it is d-detectable and there exists no other fault that generates the same observation trace.

Theorem 2 A fault $(e, f) \in F$ is d-identifiable if and only if

1. f is d-detectable,
2. for each $(e', f') \in F \setminus \{(e, f)\}$,
   (a) for each $s \in S$ and $i \in I_{\mathcal{G}}(s)$ such that $(s, f_{ori}) \notin D_B(\mathcal{G})$ and $\Lambda(i/\rho_{\mathcal{G}}(s,i)) = \Lambda(f_{in}/f_{out})$, we have $(f_{des}, \delta_{\mathcal{G}}(s,i)) \in D_F(\mathcal{G}_f, \mathcal{G}_{f'}, d)$, and
   (b) for each $s, s' \in S$, $i' \in I_{\mathcal{G}}(s')$, $i \in I_{\mathcal{G}_{f'}}(s)$, $\mu \in L(\mathcal{G}, \delta_{\mathcal{G}}(s', i'), f_{ori})$, and $\nu \in L(\mathcal{G}_{f'}, f'_{des}, s)$ such that $(s', f'_{ori}) \notin D_B(\mathcal{G})$, $\Lambda(i'/\rho_{\mathcal{G}}(s', i')) = \Lambda(f'_{in}/f'_{out})$, $\Lambda(\mu) = \Lambda(\nu)$, and $\Lambda(i/\rho_{\mathcal{G}_{f'}}(s,i)) = \Lambda(f_{in}/f_{out})$, we have $(f_{des}, \delta_{\mathcal{G}_{f'}}(s,i)) \in D_F(\mathcal{G}_f, \mathcal{G}_{f'}, d)$.

Proof: It suffices to show that Condition 2 is equivalent to the second condition in Definition 2.

Sufficiency: Pick arbitrary input/output traces $\omega \in L_f(d)$ and $\omega' \in L_{f'}(0)$. Write ω and ω' as $\omega = \mu \cdot (f_{in}/f_{out}) \cdot \nu$ and $\omega' = \mu' \cdot (f'_{in}/f'_{out}) \cdot \nu'$, where $\mu \in L(\mathcal{G}, s(0), f_{ori})$, $\nu \in L(\mathcal{G}_f, f_{des}, *)$, $|\nu| \geq d$, $\mu' \in L(\mathcal{G}, s(0), f'_{ori})$, and $\nu' \in L(\mathcal{G}_{f'}, f'_{des}, *)$.

There are two cases to consider. First, suppose that μ' can be written as $\mu' = \mu'_1 \cdot \mu'_2$, where $\Lambda(\mu) = \Lambda(\mu'_1)$. If the first input/output pair in μ'_2 is such that $\Lambda(i/\rho_{\mathcal{G}}(s,i)) \neq \Lambda(f_{in}/f_{out})$, then it's obvious that $\Lambda(\omega) \neq \Lambda(\omega')$. If the first input/output pair (i/o) in $\mu'_2 \cdot (f'_{in}/f'_{out})$ is such that $\Lambda(i/o) = \Lambda(f_{in}/f_{out})$, then by Condition 2(a), we have $\Lambda((f_{in}/f_{out}) \cdot \nu) \neq \Lambda(\mu'_2 \cdot (f'_{in}/f'_{out}) \cdot \nu')$. Therefore, we have $\Lambda(\omega) \neq \Lambda(\omega')$.

Second, suppose that ν' can be written as $\nu' = \nu'_1 \cdot \nu'_2$, where $\Lambda(\mu) = \Lambda(\mu' \cdot (f'_{in}/f'_{out}) \cdot \nu'_1)$. Then, μ can be written as $\mu_1 \cdot \mu_2$ where $\Lambda(\mu_1) = \Lambda(\mu')$. By Condition 2(b), we have $\Lambda(\mu_2 \cdot (f_{in}/f_{out}) \cdot \nu) \neq \Lambda((f'_{in}/f'_{out}) \cdot \nu')$. Therefore, we have $\Lambda(\omega) \neq \Lambda(\omega')$.

Necessity: Pick arbitrary input/output traces $\omega \in L_f(d)$ and $\omega' \in L_{f'}(0)$. Let ω and ω' be written as $\omega = \mu \cdot (f_{in}/f_{out}) \cdot \nu$ and $\omega' = \mu' \cdot (f'_{in}/f'_{out}) \cdot \nu'$ where $\mu \in L(\mathcal{G}, s(0), f_{ori})$,

$\nu \in L(\mathcal{G}_f, f_{des}, *)$, $|\nu| \geq d$, $\mu' \in L(\mathcal{G}, s(0), f'_{ori})$, and $\nu' \in L(\mathcal{G}_{f'}, f'_{des}, *)$. By assumption, we have $\Lambda(\omega) \neq \Lambda(\omega')$.

There are two cases to consider. First, suppose that $\mu' \cdot (f'_{in}/f'_{out})$ can be written as $\mu' = \mu'_1 \cdot (i/o) \cdot \mu'_2$, where $\Lambda(\mu) = \Lambda(\mu'_1)$. If the observation of input/output i/o is the same as that of faulty event f, then by assumption that $\Lambda(\omega) \neq \Lambda(\omega')$, we have $\Lambda(\nu) \neq \Lambda(\mu'_2 \cdot \nu')$. Therefore, Condition 2(a) holds.

Second, suppose that ν' can be written as $\nu' = \nu'_1 \cdot \nu'_2$, where $\Lambda(\mu) = \Lambda(\mu' \cdot (f'_{in}/f'_{out}) \cdot \nu'_1)$. Then, μ can be written as $\mu_1 \cdot (i'/o') \cdot \mu_2$ where $\Lambda(\mu_1) = \Lambda(\mu')$. If the observation of i'/o' is the same as that of faulty event f', then by assumption that $\Lambda(\omega) \neq \Lambda(\omega')$, we have $\Lambda(\nu') \neq \Lambda(\mu_2 \cdot (f_{in}/f_{out}) \cdot \nu)$. Therefore, Condition 2(b) holds.

■

3.4 Fault detection and identification algorithm

Let $Y(n) = \{(y(k) : k \in [1, n]\}$ denote the observation trace up to the nth observation, where $y(k)$ denotes the kth observation. Given $\mu \in (I \times O)^*$, let μ_l denote the last element of μ. Let $\hat{S}_F(k)$ denote the *candidate-state-fault-set*, which is the set of state–faulty event pairs that are possible after the kth observation, given by

$$\begin{aligned}\hat{S}_F(k) &= \{(s, f) : s \in S, f \in E_F \cup \{\epsilon\}, \exists \mu \in L(\mathcal{G}, s(0), f_{ori}), \nu \in L(\mathcal{G}_f, f_{des}, s), \text{ s.t.} \\ &\quad \Lambda(\mu \cdot (f_{in}/f_{out}) \cdot \nu) = Y(k), \Lambda((\mu \cdot (f_{in}/f_{out}) \cdot \nu)_l) = y(k)\}.\end{aligned}$$

The set $\hat{S}_F(k)$ is analogous to a diagnoser in [15], and consists of two kinds of state–faulty event pairs:

1. (s, ϵ), where state s is reachable by an input/output trace in $L(\mathcal{G}, s(0), s)$ that has the observation $Y(k)$ and whose last element has a non-null observation,

2. (s, f), where f is a faulty event, and state s is reachable by an input/output trace generated first in $\mathcal{G}$, and later $\mathcal{G}_f$, that has the observation $Y(k)$ and whose last element has a non-null observation.

Let $R_\epsilon(\mathcal{A}, s) = \{s' \in S : \exists \mu \in L(\mathcal{A}, s, s'), \Lambda(\mu) = \epsilon\}$ denote the *unobservable reach* of state s in $\mathcal{A}$. The set $\hat{S}_F(k)$ can be computed recursively from $\hat{S}_F(k-1)$ and the observation $y(k)$ as follows.

$$\begin{aligned}&\hat{S}_F(k) = \\ &\quad \{(\delta_{\mathcal{G}}(s_2, i), \epsilon) : \exists s_1 \in S, s_2 \in R_\epsilon(\mathcal{G}, s_1), i \in I_{\mathcal{G}}(s_2) \text{ s.t. } (s_1, \epsilon) \in \hat{S}_F(k-1), \\ &\quad \Lambda(i/\rho_{\mathcal{G}}(s_2, i)) = y(k)\} \\ &\quad \cup\{(\delta_{\mathcal{G}}(s_2, i), f) : \exists s_1 \in S, s_2 \in R_\epsilon(\mathcal{G}_f, s_1), i \in I_{\mathcal{G}_f}(s_2), \text{ s.t. } (s_1, f) \in \hat{S}_F(k-1), \\ &\quad \Lambda(i/\rho_{\mathcal{G}}(s_2, i)) = y(k)\} \\ &\quad \cup\{(\delta_{\mathcal{G}}(s_2, i), f) : \exists s_1 \in S, s_2 \in R_\epsilon(\mathcal{G}_f, f_{des}), i \in I_{\mathcal{G}_f}(s_2), \text{ s.t. } (s_1, \epsilon) \in \hat{S}_F(k-1), \\ &\quad f_{ori} \in R_\epsilon(\mathcal{G}, s_1), \Lambda(e) = y(k)\}.\end{aligned}$$

We are now ready to give an algorithm for on-line fault detection and identification. The following algorithm detects the occurrence of a fault, and if possible, identifies the fault.

Algorithm 2 *On-line fault detection and identification algorithm*

```
  $\hat{S} \leftarrow \{s(0)\}$
  Loop: if NEW-OBSERVATION
      input($y$)
      update($\hat{S}$)
      if $\nexists s \in S$ such that $(s, \epsilon) \in \hat{S}$ /* fault detection */
          $\hat{F} \leftarrow \{f \in F : \exists s \in S$ such that $(s, f) \in \hat{S}\}$
          if some $f \in \hat{F}$ is identifiable and $|\hat{F}| \neq 1$
              goto Loop
          else
              return $\hat{F}$
  else
      goto Loop
end
```

The algorithm above continually checks for the presence of a new observation. Whenever a new observation is received, the algorithm updates the candidate-state-fault-set $\hat{S}$, and examines whether or not a faulty event occurred. A fault is detected when the candidate-state-fault-set contains only pairs that do not have ϵ as their second elements. A fault is identified when the candidate-state-fault-set consists of only one element and the element has an associated faulty event (not ϵ).

4 Conclusions

In this paper, we considered the problem of detectability and identifiability of faults with a given delay in communication protocols, modeled by a finite-state machine. We considered faults specified as addition of events (arcs), as well as the four common faults considered in the conformance testing of communication protocols. We gave structural conditions that are necessary and sufficient for detectability and identifiability of a fault with a given delay, and provided an algorithm for on-line fault detection and identification. Our results also characterize the smallest delay for a system to be detectable and identifiable. Specifically, given a nominal FSM $\mathcal{G}$, a set of specified faults F, and an observation map Λ, we can use the conditions in Theorems 1 and 2 to compute the the smallest value of d such that each fault in F is d-detectable (or d-identifiable). These delay bounds can be taken into account in the design of on-line monitoring systems [17].

For future research, detectability and identifiability of multiple faults are of interest. Incorporating timing information to fault detection is also a possible research direction.

References

[1] S. Bavishi and E. K. P. Chong, "Testability analysis using a discrete event systems framework," in *Proc. of 1995 American Control Conference*, (Seattle, Washington, USA), pp. 2621–2625, June 1995.

[2] A. Bouloutas, G. W. Hart, and M. Schwartz, "Fault identification using a finite state machine model with unreliable partially observed data sequences," *IEEE Trans. Communications*, vol. 41, no. 7, pp. 1074–1083, July 1993.

[3] A. Bouloutas, G. W. Hart, and M. Schwartz, "On the design of observers for fault detection in communication networks," in *Network Management and Control* (A. Kershenbaum, M. Malek, and M. Wall, eds.), (New York), pp. 319–338, Plenum Press, 1990.

[4] A. Bouloutas, G. W. Hart, and M. Schwartz, "Simple finite-state fault detectors for communication networks," *IEEE Trans. Communications*, vol. 40, no. 3, pp. 477–479, Mar. 1992.

[5] L. E. Holloway and S. Chand, "Time templates for discrete event fault monitoring in manufacturing systems," in *Proc. of 1994 American Control Conference*, pp. 701–706, June 1994.

[6] J. Hopcroft and J. Ullman, *Introduction to Automata Theory, Languages, and Computation.* Reading, MA, USA: Addison-Wesley, 1979.

[7] F. Lin, "Diagnosability of discrete event systems and its applications," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 4, no. 2, pp. 197–212, May 1994.

[8] R. E. Miller and S. Paul, "Structural analysis of protocol specifications and generations of maximal fault coverage conformance test sequences," *IEEE/ACM Trans. Networking*, vol. 2, no. 5, pp. 457–470, Oct. 1994.

[9] C. M. Özveren and A. S. Willsky, "Invertibility of discrete-event dynamic systems," *Mathematics of Control, Signals, and Systems*, vol. 5, pp. 365–390, 1992.

[10] C. M. Özveren and A. S. Willsky, "Observability of discrete event systems," *IEEE Trans. Automatic Control*, vol. 35, no. 7, pp. 797–806, July 1990.

[11] Y. Park and E. K. P. Chong, "On *d*-inversion in timed discrete event systems with interruption," in *Proc. of 34th Conf. Decision and Control*, (New Orleans, Louisiana, USA), 1995. To appear.

[12] Y. Park and E. K. P. Chong, "Distributed inversion in timed discrete event systems," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 5, no. 2/3, pp. 219–241, April/July 1995.

[13] J. Prock, "A new technique for fault detection using Petri nets," *Automatica*, vol. 27, pp. 239–245, 1991.

[14] P. Ramadge, "Observability of discrete event systems," in *Proc. of 25th Conf. Decision and Control*, (Athens, Greece), pp. 1108–1112, Dec. 1986.

[15] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Diagnosability of discrete-event systems," *IEEE Trans. Automatic Control*, vol. 40, no. 9, pp. 1555–1575, Sept. 1995.

[16] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Failure diagnosis using discrete event models," Tech. Rep. # CGR-94-03, Control Group, College of Engineering, University of Michigan, USA, May 1994.

[17] B. A. Schroeder, "On-line monitoring: A tutorial," *IEEE Computer*, pp. 72–78, June 1995.

On Observability and Nondeterminism in Discrete Event Control [1]

Michael Heymann[1] and Feng Lin[2]
[1]Department of Computer Science
Technion, Israel Institute of Technology, Haifa 32000, Israel
[2]Department of Electrical and Computer Engineering
Wayne State University, Detroit, MI 48202, USA

Abstract

The control of partially observed discrete event systems has received a great deal of attention in the literature. Necessary and sufficient conditions for existence of supervisors, algorithms for supervisor synthesis, for off-line as well as on-line implementation, have been obtained, and a wide variety of important questions have been investigated. In recent years, a related problem has been receiving increasing attention in the literature; that is, the supervisory control problem of nondeterministic discrete event systems. While it is widely understood that partial observation may lead to nondeterminism, the theory of nondeterministic supervisory control has evolved along a completely different avenue than the theory of partially observed systems. Also, while some existence conditions for control of nondeterministic systems have been derived, few explicit algorithms for supervisor synthesis have been obtained. More importantly, the precise connections between the two frameworks have never been examined in detail. In the present paper we investigate the problem of supervisory control of nondeterministic discrete event systems from the point of view of control under partial observation. We carefully examine the relation between the two frameworks and we derive algorithms for supervisor synthesis for nondeterministic systems under a wide variety of conditions. In the present paper we examine nondeterministic systems with both deterministic and nondeterministic specifications in the closed case. The general (nonclosed) case will be discussed elsewhere [4].

1 Introduction

Since the introduction of observability [8], a great deal of work has been done on the control of partially observed discrete event systems: Necessary and sufficient conditions for existence of supervisors, algorithms for supervisor synthesis, for off-line as well as on-line implementation, have been obtained, and a wide variety of have been investigated. In recent years a related problem has also been receiving increasing attention; that is, the supervisory control problem of nondeterministic discrete event systems. In this setting, a system is given either as a nondeterministic automaton (with ϵ-transitions) or as a *trajectory model* [1] [2] (see also [6] [7] [10] [12]). The specification of legal behavior of such systems can be given either in a deterministic or in a nondeterministics setting. The supervisory control problem is to synthesize a supervisor that restricts the nondeterministic plant so as to satisfy the corresponding specification.

While it is widely understood that partial observation may lead to nondeterminism, the theory of nondeterministic supervisory control has evolved along an independent and completely different avenue than the theory of partially observed systems [12] [6]. Also, while some existence conditions for control of nondeterministic systems have been derived, few explicit algorithms for supervisor synthesis have been obtained. More importantly, the precise connections between the two frameworks have never been examined in detail.

---

[1] This research is supported in part by the National Science Foundation under grant ECS-9315344 and in part by the Technion Fund for Promotion of Research.

It is intuitively clear that if one were given a deterministic model, a deterministic specification of legal behavior, and a partial observation map, one could design a supervisor that is at least as powerful as the corresponding supervisor that is based only on knowledge of the projected plant and specification. This is because the nondeterministic (projected) model (and specification) can be derived from the deterministic model (and specification) while the converse is not generally true. Indeed it can be shown, that under certain circumstances, the deterministic partial observation setting is strictly more powerful that the nondeterministic counterpart. Conversely, one may be given a nondeterministic model, along with a deterministic or nondeterministic specification, as the basis for synthesis of a supervisor. One can then try to attack the problem directly in the nondeterministic setting or, alternatively, one can view the problem as a consequence of a deterministic framework that appears to be nondeterministic because of the internalization (unobservability) of some hypothetical events. In the latter case, the supervisor could have been designed by the conventional framework of control under partial observation. A natural question that arises is whether the two approaches lead to the same results. It is the focus of the present paper to investigate in detail the above raised issues.

We first briefly review the theory of supervisory control of discrete event systems under partial observation and recall the main relevant results on the existence of supervisors and some common procedures for supervisor synthesis. We also review in some detail the main concepts of nondeterministic discrete event systems and their representations, and reexamine the relation between the trajectory models and their corresponding nondeterministic automata. It is then shown that if the language describing a deterministic system is Σ^*-observable, then the projection of the system is deterministic. On the other hand, if the language is unobservable, its projection is not necessarily nondeterministic as can be shown by example. However, we prove that if all trajectories of the projected system remain "valid", then the projected system is nondeterministic.

Next, a "lifting" formalism is derived by which the nondeterministic system is translated (or lifted) to a certain deterministic system by introducing hypothetical events (that are assumed to be the internalized events whose unobservability led to nondeterminism). The lifted system is constructed so that its projection yields back the original nondeterministic system. It is then easily seen that the language obtained by the lifting of any language that is defined over the original event set to the extended event set of the lifted system, is normal with respect to the lifted system. This fact immediately implies that there is a very natural correspondence between optimal supervisors obtained for the lifted system to satisfy a normal language specification under partial observation, and supervisors for the nondeterministic system designed to satisfy deterministic (language) specifications. Specifically, a necessary and sufficient condition for satisfying a language specification by supervisory control of a nondeterministic system, is controllability of the language. This result has been previously obtained in [12] via a supervisory control theory of nondeterministic systems. While the direct derivation of this result in the nondeterministic framework is far from being trivial, it is straightforward by the above discuss. In the present paper we examine nondeterministic systems with both deterministic and nondeterministic specifications in the closed case. The general (nonclosed) case will be discussed elsewhere [4].

2 Supervisory control under partial observation

Let us now review the basic results of supervisory control for deterministic systems under partial observation. The uncontrolled system is described by a (deterministic) automaton $G = (\Sigma, Q, \delta, q_o, Q_m)$ with its elements defined in a usual way. The languages generated and marked by G are denoted by $L(G)$ and $L_m(G)$ respectively. The event set is partitioned into controllable (observable) and uncontrollable (unobservable) event sets: $\Sigma = \Sigma_c \cup \Sigma_{uc}(=$

$\Sigma_o \cup \Sigma_{uo}$). A supervisor is a disablement map $\gamma : PL(G) \to 2^{\Sigma_c}$ (where $P : \Sigma^* \to \Sigma_o^*$ is the projection that deletes the unobserved events) such that following an observed trace $s \in PL(G)$, the events $\sigma \in \gamma(s)$ are disabled by the supervisor. The languages generated and marked by the supervised system are denoted by $L(\gamma/G)$ and $L_m(\gamma/G) = L(\gamma/G) \cap L_m(G)$ respectively, where $L(\gamma/G)$ is given recursively by

1 $\epsilon \in L(\gamma/G)$

2 $(\forall s \in L(\gamma/G))(\forall \sigma \in \Sigma) s\sigma \in L(\gamma/G) \Leftrightarrow s\sigma \in L(G) \wedge s \in L(\gamma/G) \wedge \sigma \notin \gamma(s)$.

We say K is closed if it equals its prefix closure $K = \overline{K}$ and K is $L_m(G)$-closed if $K = L_m(G) \cap \overline{K}$. We also define controllability and observability as follows [11] [8].

Definition 1 A language $K \subseteq L(G)$ is controllable (with respect to Σ_{uc}) if

$$(\forall s \in \overline{K})(\forall \sigma \in \Sigma_{uc}) s\sigma \in L(G) \Rightarrow s\sigma \in \overline{K}.$$

Definition 2 A language $K \subseteq L(G)$ is observable (with respect to Δ and Σ_{uo}) if

$$(\forall s, s' \in \overline{K}) Ps = Ps' \Rightarrow (\forall \sigma \in \Delta)(s\sigma \in \overline{K} \wedge s'\sigma \in L(G) \Rightarrow s'\sigma \in \overline{K}).$$

The controllability and observability characterize the existence of a supervisor as proved in [8] and summarized in the following theorem.

Theorem 1 Let $K \subseteq L(G)$ be non-empty. Then there exists a supervisor γ such that $L(\gamma/G) = K$ if and only if K is closed, controllable and observable.

Another useful concept is normality which is defined as follows [8].

Definition 3 A language $K \subseteq L(G)$ is normal if

$$(\forall s \in L(G)) Ps \in P\overline{K} \Rightarrow s \in \overline{K}.$$

One important fact regarding the relation between observability and normality is the following theorem, which is basic to our development in this paper [9].

Theorem 2 Assume $\Sigma_c \subseteq \Sigma_o$. Then a language is controllable and observable if and only if it is controllable and normal.

A nice property of controllable (normal) languages is that they are closed under union. So let $\mathcal{C}(E)$ ($\mathcal{N}(E)$, $\mathcal{CN}(E)$, respectively) denote the set of controllable (normal, controllable and normal, respectively) sublangauges of E. then we have [8] [9].

Theorem 3 The supremal elements $sup\mathcal{C}(E)$, $sup\mathcal{N}(E)$, and $sup\mathcal{CN}(E)$ exist and they preserve the properties of closeness and $L_m(G)$-closeness.

3 Nondeterminism

In this section we briefly review the trajectory-model formalism of [2] (see also [12]) that has been developed as the basic tool for modeling and analysis of nondeterministic discrete event systems.

Just as the *trace* $s \in L(G) \subseteq \Sigma^*$ is a record of the string of events executed in a given run of a system G, the *trajectory* is also a record associated with a run of G. It is more detailed than the trace in that it lists, in addition to the successfully executed events, also events that the system might have rejected (or refused), if offered, after each successful event. Thus, a trajectory is an oject in $2^\Sigma \times (\Sigma \times 2^\Sigma)^*$ of the form

$$t = (X_0, \sigma_1, X_1, ..., X_{k-1}, \sigma_k, X_k)$$

where σ_i denotes the ith executed event, and X_i, the ith *refusal*, denotes the set of events refused after the ith executed event. The *initial refusal* X_0 is the set of events that are refused before any event is executed. We call the integer k the *length* of t, denoted $|t|$, and the trace associated with t is defined as

$$tr(t) = \sigma_1 ... \sigma_k.$$

A trajectory is called *valid* if for all $i > 0$, $\sigma_i \notin X_{i-1}$ (that is, an event cannot be executed if it has just been refused).

Let t be a trajectory given by

$$t = (X_0, (\sigma_1, X_1) ... (\sigma_k, X_k)).$$

A trajectory r is a *prefix* of t, denoted $r \preceq t$, if

$$r = (X_0, (\sigma_1, X_1) ... (\sigma_j, X_j))$$

and $0 \leq j \leq k$. The trajectory r is called a *proper* prefix of t if $j < k$. We denote the prefix of length j of t by $pref_j(t)$ (so that $pref_0(t) = (X_0, \epsilon)$, where ϵ denotes the empty string, and $pref_k(t) = t$). The set of all prefixes of t is called the *prefix-closure* of t and is denoted $pref(t)$.

A trajectory r is said to be *dominated* by t, denoted $r \sqsubseteq t$, if it is of the form

$$r = (Y_0, (\mu_1, Y_1) ... (\mu_k, Y_k)),$$

with $\mu_i = \sigma_i$ for $1 \leq i \leq k$ and $Y_j \subseteq X_j$ for $0 \leq j \leq k$. The set of all trajectories dominated by t is called the *completion*, or *dominance-closure*, of t and denoted $comp(t)$.

Finally, we define the *closure* of t, denoted $cl(t)$, as

$$cl(t) := \bigcup_{v \in comp(t)} pref(v)$$

and the closure of a set of trajectories $\mathcal{T}$, is given by

$$cl(\mathcal{T}) := \bigcup_{t \in \mathcal{T}} cl(t).$$

A set of trajectories $\mathcal{T}$ is *closed* [2] if

$$\mathcal{T} = cl(\mathcal{T}).$$

We say that a set of trajectories $\mathcal{T}$ is *saturated*[3] if the following condition holds:

$$\begin{aligned} &(\forall k = 1, 2, ...)(\forall j : 0 \leq j \leq k)(\forall \sigma \in \Sigma - X_j) \\ &(((X_0, (\sigma_1, X_1) ... (\sigma_k, X_k)) \in \mathcal{T} \wedge (X_0, (\sigma_1, X_1) ... (\sigma_j, X_j)(\sigma, \emptyset)) \notin \mathcal{T}) \\ &\Rightarrow (X_0, (\sigma_1, X_1) ... (\sigma_j, X_j \cup \{\sigma\}) ... (\sigma_k, X_k)) \in \mathcal{T}). \end{aligned}$$

We are now in a position to define a (nondeterministic) process through its associated set of trajectories. Intuitively, we identify a process $\mathcal{P}$ with the set of all trajectories associated with possible runs of $\mathcal{P}$. More formally, we have the following

---

[2] A closed set of trajectories is always nonempty since it includes the null trajectory $(\emptyset, \epsilon)$.

[3] Note that the term "saturated" as defined here is different from the way it was used in [12].

Definition 4 [4] A (possibly) nondeterministic *process* $\mathcal{P}$ is a closed and saturated subset $\mathcal{P} \subseteq 2^{\Sigma} \times (\Sigma \times 2^{\Sigma})^{*}$ *of valid trajectories.*

The saturation condition on the set of trajectories of a process implies that if an event is impossible it will be refused. (We shall later see that while in nondeterministic processes events need not be impossible to be refused, in deterministic processes events are refused if and only if they are impossible.)

Let $\mathcal{T}$ be a set of trajectories. We say that a trajectory $t \in \mathcal{T}$ is *dominant* in $\mathcal{T}$ if there is no trajectory $t' \in \mathcal{T}$, $t' \neq t$, such that $t \sqsubseteq t'$. The set of all trajectories that are dominant in $\mathcal{T}$ is called the *dominance set* of $\mathcal{T}$ and is denoted $dom(\mathcal{T})$. A trajectory in $\mathcal{T}$ is called *maximal* if it is dominant and there is no trajectory $t' \in \mathcal{T}$ such that $t \preceq t'$. The subset of all maximal trajectories in $\mathcal{T}$ is called the *maximal set* of $\mathcal{T}$ and is denoted $max(\mathcal{T})$.

The following Theorem states that a process $\mathcal{P}$ is completely characterized by its dominance set.

Theorem 4 Let $\mathcal{P}$ be a process. Then $cl(dom(\mathcal{P})) = \mathcal{P}$.

We are now ready to define a *deterministic* process in the trajectory model setting:

Definition 5 A process $\mathcal{P}$ is called *deterministic* if for every trajectory $(X_0, (\sigma_1, X_1)...(\sigma_k, X_k)) \in \mathcal{P}$ and any $\sigma \in \Sigma$

$$(X_0, (\sigma_1, X_1)...(\sigma_k, X_k)(\sigma, \emptyset)) \in \mathcal{P} \Leftrightarrow (X_0, (\sigma_1, X_1)...(\sigma_k, X_k \cup \{\sigma\})) \notin \mathcal{P}.$$

Thus, a process is deterministic whenever events are refused if and only if they are impossible.

Now, let $L \subseteq \Sigma^{*}$ be a prefix-closed language (set of traces) and consider the class of all trajectory models that share L as their trace set. We note that this set is never empty and construct the following trajectory model, denoted $det(L)$, that belongs to this class:

Algorithm 1 *Construction of $det(L)$.*

1. $(\emptyset, \epsilon) \in det(L)$.
2. Proceed by induction on string length: For $t = (X_0, (\sigma_1, X_1)...(\sigma_k, X_k)) \in det(L)$ and $\sigma \in \Sigma$,

$$\begin{aligned} &(X_0, (\sigma_1, X_1)...(\sigma_k, X_k \cup \{\sigma\})) \in det(L) \Leftrightarrow tr(t)^\frown \sigma \notin det(L) \\ &(X_0, (\sigma_1, X_1)...(\sigma_k, X_k)(\sigma, \emptyset)) \in det(L) \Leftrightarrow tr(t)^\frown \sigma \in det(L) \end{aligned}$$

The following theorem summarizes our preceding discussion and characterizes deterministic processes [2] [12]:

Theorem 5 Let $\mathcal{P}$ be a process and let $L(\mathcal{P})$ be its trace set. Then $\mathcal{P}$ is deterministic if and only if for every process $\mathcal{Q}$ such that $L(\mathcal{Q}) = L(\mathcal{P})$,

$$\mathcal{P} = det(L(\mathcal{P})) \subseteq \mathcal{Q}.$$

Thus a deterministic process is uniquely defined by its associated trace set and, in fact, is the smallest process associated with a given trace set.

In view of the preceding discussion it is clear that if $\mathcal{P}$ and $\mathcal{Q}$ are two processes defined over the same event set Σ, we are justified in saying that $\mathcal{P}$ is *more nondeterministic* than $\mathcal{Q}$

---

[4] The above definition is a simplified version of Definition 12.1 in [2] since we deal here only with termination-free nondivergent processes. The concepts of termination and divergence are discussed in detail in [2]. Intuitively, a process is divergent if it can engage in an unbounded string of unobserved transitions.

whenever $\mathcal{Q} \subseteq \mathcal{P}$. (In particular, it is shown in [2] that $\mathcal{P}$ can then be thought of as evolving into $\mathcal{Q}$ through ϵ–transitions).

For each trajectory model, we can construct a corresponding nondeterministic automaton with ϵ-transitions, using the algorithm in [2]. Similarly, for each nondeterministic automaton with ϵ-transitions, it is easy to construct a corresponding trajectory model by first identifying for each state a *maximal-refusal-set* and then associating with each ϵ–stable path [4] a corresponding dominant trajectory. Therefore, we can use either the trajectory model or its associated nondeterministic automaton to model a nondeterministic system. Henceforth, in this paper, we shall use the same symbol $\mathcal{R}$ to denote both the nondeterministic automaton and the trajectory model of a process. The languages generated and marked by $\mathcal{R}$ are denoted by $L(\mathcal{R})$ and $L_m(\mathcal{R})$ respectively.

4 Nondeterminism and unobservability

Let $\tilde{\mathcal{R}}$ be a process over the event set $\Sigma \cup \Sigma'$, where Σ' is the set of unobserved internal events. In other words, an external observer (e.g., a supervisor) of $\tilde{\mathcal{R}}$ can observe the events in Σ but not events in Σ'. To obtain a suitable model for the partially observed process, we define a projection (or *internalization*) operator $\pi_{\Sigma'} : 2^{\Sigma \cup \Sigma'} \times ((\Sigma \cup \Sigma') \times 2^{\Sigma \cup \Sigma'})^* \to 2^{\Sigma} \times (\Sigma \times 2^{\Sigma})^*$ as follows.

For a trajectory

$$t = (X_o, (\sigma_1, X_1), ..., (\sigma_k, X_k))$$

the projection $t\backslash_{\Sigma'} = \pi_{\Sigma'}(t)$ is obtained from t by the following procedure.

1. Delete from t all occurrences of event symbols that belong to Σ' (both as executed events and refused events). Thus, each refusal set X_i becomes $X_i - \Sigma'$.

2. Replace all consecutive refusal sets whose associated execution event symbols have been deleted, by their union. That is, if (in t) $\sigma_{i-1} \in \Sigma$ and $\sigma_i, \sigma_{i+1}, ..., \sigma_l \in \Sigma'$, then replace $X_{i-1} - \Sigma'$ by $(X_{i-1} - \Sigma') \cup (X_i - \Sigma') \cup ... \cup (X_l - \Sigma')$.

In general, the trajectory $t\backslash_{\Sigma'}$ thus obtained may not be valid. (Recall that a trajectory is valid if $\sigma_j \notin X_{j-1}$ for all j). The projection $\tilde{\mathcal{R}}\backslash_{\Sigma'}$ of a process $\tilde{\mathcal{R}}$ is a obtained as a subset of the set of valid trajectories projected from $\tilde{\mathcal{R}}$, as defined in [2] [4][5]. An alternate way to obtain the projection is to view $\mathcal{R}$ as a nondeterministic automaton, in which case $\tilde{\mathcal{R}}\backslash_{\Sigma'}$ is obtained from $\mathcal{R}$ by replacing in it all events $\sigma \in \Sigma'$ by ϵ.

Let $\tilde{\mathcal{R}}$ be a deterministic process over $\Sigma \cup \Sigma'$. We define the observability of $\tilde{\mathcal{R}}$ in terms of $L(G) = (\Sigma \cup \Sigma')^*$, $\Delta = \Sigma$ and $\Sigma_{uo} = \Sigma'$, that is, $\tilde{\mathcal{R}}$ is observable if

$$(\forall s, s' \in L(\tilde{\mathcal{R}}))Ps = Ps' \Rightarrow (\forall \sigma \in \Sigma)(s\sigma \in L(\tilde{\mathcal{R}}) \Rightarrow s'\sigma \in L(\tilde{\mathcal{R}}))$$

where P is the projection from $(\Sigma \cup \Sigma')^*$ to Σ^*.

Let $\mathcal{R}$ be the process obtained by projecting $\tilde{\mathcal{R}}$ on Σ; that is, $\mathcal{R} = \tilde{\mathcal{R}}\backslash_{\Sigma'}$. Then the observability of $\tilde{\mathcal{R}}$ is related to the determinism of $\mathcal{R}$ as shown in the following two theorems.

Theorem 6 If $\tilde{\mathcal{R}}$ is observable, then $\mathcal{R}$ is deterministic.

Theorem 7 If $\tilde{\mathcal{R}}$ is unobservable and every trajectory obtained from its projection is valid, then $\mathcal{R}$ is nondeterministic.

---

[5] The detailed definition of the projection of a process is omitted here because of space limitation.

Example 1 The projection of the process $\mathcal{R} = cl(\{b\}, a, \{a\}, b, \{a, b\})$ on $\{b\}$ is $\mathcal{R}\backslash_{\{a\}} = cl(\emptyset, b, \{b\})$ which is deterministic. Note, however, that the projection of the trajectory $t = (\{b\}, a, \{a\}, b, \{a, b\})$ is $t\backslash_{\{a\}} = (\{b\}, b, \{b\})$ which is not valid.

Consider a nondeterministic automaton, possibly with $\epsilon-$transitions,

$$\mathcal{R} = (\Sigma \cup \{\epsilon\}, Q, \delta, q_o)$$

over the event set Σ. We introduce now a procedure for constructing a deterministic automaton

$$\tilde{\mathcal{R}} = (\Sigma \cup \Sigma', \tilde{Q}, \tilde{\delta}, q_o)$$

over an extended event set $\Sigma \cup \Sigma'$, such that $\mathcal{R} = \tilde{\mathcal{R}}\backslash_{\Sigma'}$. That is, $\tilde{\mathcal{R}}$ reduces to $\mathcal{R}$ upon replacing by ϵ-transitions all its transitions labeled by events in Σ'. The procedure is based on first extending $\mathcal{R}$ to a standard nondeterministic automaton with $\epsilon-$transitions, and then replacing the ϵ labels by labels from the event set $\Sigma' = \{\tau_1, \tau_2, ...\}$.

Procedure Extend($\mathcal{R} \rightarrow \tilde{\mathcal{R}}$)

1. $\tilde{Q} := Q$;

2. For each $q \in \tilde{Q}$ and $\sigma \in \Sigma$: If $|\delta(q, \sigma)| > 1$, add one more state, q' and add $\epsilon-$transitions as follows:

   $$\tilde{Q} := \tilde{Q} \cup \{q'\};\ \tilde{\delta}(q, \sigma) := \{q'\};\ \tilde{\delta}(q', \epsilon) := \delta(q, \sigma);$$

   else set

   $$\tilde{\delta}(q, \sigma) := \delta(q, \sigma);$$

3. For each $q \in \tilde{Q}$: replace the $\epsilon-$transitions by transitions labeled $\tau_1, \tau_2, ...$ as follows: If $\tilde{\delta}(q, \epsilon) = \{q_1, ..., q_n\}$, then set

   $$\tilde{\delta}(q, \tau_1) := \{q_1\};\ ...;\ \tilde{\delta}(q, \tau_n) := \{q_n\};$$

5 Supervisory control: closed case

In this section, we consider the supervisory control of closed nondeterministic systems, that is, we assume every state is marked and we do not consider the issue of blocking. The general (nonclosed) case has also been investigated and will be considered in [4].

We allow both the system and the specification to be nondeterministic, that is, to be modeled by trajectory models or nondeterministic automata. We shall denote the system and specification by $\mathcal{R}$ and $\mathcal{H}$, respectively, where the nondeterministic automaton of $\mathcal{H}$ is given by

$$\mathcal{H} = (\Sigma \cup \{\epsilon\}, P, \mu, p_0).$$

Naturally, we shall require that $\emptyset \neq L(\mathcal{H}) \subseteq L(\mathcal{R})$.

Control is achieved as usual by a supervisor $\gamma : L(\mathcal{R}) \rightarrow 2^{\Sigma_c}$. The supervised system is denoted by $\gamma/\mathcal{R}$. To formally obtain $\gamma/\mathcal{R}$, we first compute the language $L(\gamma/\mathcal{R})$ as defined in Section 2. The behavior of the supervised system is then described by

$$\gamma/\mathcal{R} = \mathcal{R}||det(L(\gamma/\mathcal{R})),$$

where $||$ denotes the synchronous composition.

Our goal is to synthesize a supervisor γ such that, if possible,

$$\gamma/\mathcal{R} = \mathcal{H}.$$

To obtain a necessary and sufficient condition for the existence of such a supervisor, we proceed as follows. First, we extend $\mathcal{H}$ to

$$\overline{\mathcal{H}} := (\Sigma \cup \{\epsilon\}, P \cup \{b\}, \overline{\mu}, p_0, P),$$

where the state b is called the "bad" state and the transition function $\overline{\mu}$ is given by

$$\overline{\mu}(p,\sigma) := \begin{cases} \mu(p,\sigma) & \text{if } \mu(p,\sigma) \text{ is defined,} \\ b & \text{if } \mu(p,\sigma) \text{ is undefined.} \end{cases}$$

Note that $L(\overline{\mathcal{H}}) = \Sigma^*$ and $L_m(\overline{\mathcal{H}}) = L(\mathcal{H})$.

Next, we take the synchronous composition of $\mathcal{R}$ and $\overline{\mathcal{H}}$ to obtain:

$$\mathcal{F} = Acc(\mathcal{R}||\overline{\mathcal{H}}) = Acc(\Sigma \cup \{\epsilon\}, F, \delta, f_o, F_b).$$

where $Acc(.)$ denotes the accessible component. The state set $F = Q\times(P\cup\{b\})$ is partitioned into two classes: "good" states $F_g = Q\times P$ and "bad" states $F_b = Q\times\{b\}$. It is readily noted that, since $L(\overline{\mathcal{H}}) = \Sigma^*$, it follows that $L(\mathcal{F}) = L(\mathcal{R})$. Thus, $\mathcal{F}$ constitutes a model of the process, and the specification is modeled as the subprocess of $\mathcal{F}$ that consists of all trajectories whose corresponding states do not intercept F_b. In some applications, the system may be given directly as a nondeterministic process (or automaton) $\mathcal{F}$ with the specification as a sub-process. In this latter case the above procedures for obtaining $\mathcal{F}$ can be omitted.

We next use the procedure Extend, given in the previous section, to "lift" $\mathcal{F}$ to a deterministic automaton $\tilde{\mathcal{F}}$ by adding a set Σ' of unobservable events. The set of good states $\tilde{F}_g$ and bad states $\tilde{F}_b$ in $\tilde{\mathcal{F}}$ are defined accordingly. Thus, we obtain

$$\tilde{\mathcal{F}} = (\Sigma \cup \Sigma', \tilde{F}, \tilde{\delta}, f_o, \tilde{F}_b),$$

where

$$\tilde{F}_b = F_b \cup ((\tilde{F} - F) \cap \{\tilde{\delta}(q,\sigma) : |\delta(q,\sigma)| > 1 \wedge \delta(q,\sigma) \cap (F - F_b) = \emptyset.\})$$

Define the "legal" language $E \subseteq L(\tilde{\mathcal{F}})$ to be the set of all traces that visit only good states in $\tilde{\mathcal{F}}$:

$$E = \{s \in L(\tilde{\mathcal{F}}) : (\forall t \preceq s)\tilde{\delta}(x_o, t) \notin \tilde{F}_b\}.$$

Clearly, $det(E)\backslash_{\Sigma'} = \mathcal{H}$.

With these preparations, we can now state a necessary and sufficient condition for the existence of a supervisor where in the following theorem, observability is defined for $\Delta = \Sigma \cup \Sigma'$ and $\Sigma_{uo} = \Sigma'$.

Theorem 8 For $\mathcal{R}$ and $\mathcal{H}$ given as above, there exists a supervisor γ such that $\gamma/\mathcal{R} = \mathcal{H}$ if and only if E is controllable and observable with respect to $L(\tilde{\mathcal{F}})$.

The above theorem shows that we can translate a supervisory control problem for a nondeterministic system into a supervisory control problem for an lifted deterministic system: We can first synthesize $\tilde{\gamma}$ for the deterministic system and then (easily) obtain γ from $\tilde{\gamma}$ for the nondeterministic system by letting $\gamma(s) = \tilde{\gamma}(s) \cap \Sigma$. The significance of this result is that all the methodology developed for deterministic systems can now be applied to nondeterministic systems. In particular, if E is not controllable and observable, then the best we can do is to design a supervisor using the following

Algorithm 2 *Off-line synthesis*

1. Extend($\mathcal{F} \to \tilde{\mathcal{F}}$);
2. $SupCN(\tilde{\mathcal{F}} \to \tilde{\mathcal{F}}_{CN})$;
3. Design a supervisor off-line.

In Algorithm 2, Step 1 was described earlier. Step 2 calculates the supremal controllable and normal sublanguage of E [8]. We note that since in $\tilde{\mathcal{F}}$, all controllable events are observable ($\Sigma_c \subseteq \Sigma = \Sigma_o$), controllability and observability of a language is equivalent to controllability and normality of that language. In step 3 a supervisor is designed off-line in the usual way [8]. The supervisor thus obtained is "optimal" in the sense that it generates the largest behavior possible as shown in the following theorem.

Theorem 9 The supervisor designed using Algorithm 2 is an optimal supervisor.

Two other approaches can be used for supervisor synthesis. The second approach is to design a supervisor on-line instead of off-line. The advantage of this approach is that the computational complexity is linear at each step of event execution.

Algorithm 3 *On-line synthesis*

1. Extend($\mathcal{F} \to \tilde{\mathcal{F}}$);
2. $SupC(\tilde{\mathcal{F}} \to \tilde{\mathcal{F}}_C)$;
3. Design a supervisor on-line.

Step 1 is same as that of Algorithm 2. Step 2 calculates the supremal controllable sublanguage of the legal language E. This can be done with linear complexity for a closed language E. In Step 3, we design a supervisor on-line using the results of [3]. The resulting supervisor will generate the supremal controllable and normal sublanguage of E, and hence is also optimal. As shown in [3], the complexity at each step of event execution is linear.

In view of the way $SupCN$ is calculated, we can modify Algorithm 2 by directly converting $\mathcal{F}$ to a deterministic automaton $\overline{\mathcal{F}}$ without adding the unobservable events Σ'. This leads to our third approach.

Algorithm 4 *Direct synthesis*

1. Convert($\mathcal{F} \to \overline{\mathcal{F}}$);
2. $SupC(\overline{\mathcal{F}} \to \overline{\mathcal{F}}_C)$;
3. Design a supervisor off-line.

Procedure Convert that converts the nondeterministic automaton $\mathcal{F}$ into a deterministic one $\overline{\mathcal{F}}$ is standarded [5]. Each state in $\overline{\mathcal{F}}$ is now a subset of states in $\mathcal{F}$. We call such a state "bad" if it contains a bad state of $\mathcal{F}$. Step 2 computes the supremal controllable sublanguage.

We can show that the resulting supervised systems using Algorithm 2 and Algorithm 4 are the same as far as traces in Σ^* are concerned.

Theorem 10 Let γ_1 and γ_3 be the supervisors obtained from Algorithm 2 and Algorithm 4, respectively. Then

$$L(\gamma_1/\mathcal{R}) = L(\gamma_3/\mathcal{R}).$$

The above results and algorithms can be generalized to non-closed case and will be discussed in [4].

References

[1] M. Heymann, 1990. Concurrency and discrete event control. *IEEE Control Systems Magazine, 10(4)*, pp. 103-112.

[2] M. Heymann and G. Meyer, 1991. An algebra of discrete event processes. *NASA Technical Memorandum 102848.*

[3] M. Heymann and F. Lin, 1994. On-line control of partially observed discrete event systems. *Discrete Event Dynamic Systems: Theory and Applications, 4(3)*, pp. 221-236.

[4] M. Heymann and F. Lin, 1995. On observability and nondeterminism in discrete event systems: the general case. *forthcoming.*

[5] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley, 1979.

[6] R. Kumar and M. A. Shayman, 1993. Non-blocking supervisory control of nondeterministic systems via prioritized synchronization. *Technical Research Report, T.R. 93-58*, Institute for Systems Research, University of Maryland.

[7] R. Kumar and M. A. Shayman, 1994. Supervisory control under partial observation of nondeterministic systems via prioritized synchronization. *Technical Report,* Department of Electrical Engineering, University of Kentucky.

[8] F. Lin and W. M. Wonham, 1988. On observability of discrete event systems. *Information Sciences, 44(3)*, pp. 173-198.

[9] F. Lin and W. M. Wonham, 1994. Supervisory control of timed discrete event systems under partial observation, *IEEE Transactions on Automatic Control, 40(3)*, pp. 558-562.

10] A. Overkamp, 1994. Supervisory control for nondeterministic systems. *Proccedings of 11th International Conference on Analysis and Optimization of Systems*, pp. 59-65.

11] R. J. Ramadge and W. M. Wonham, 1987. Supervisory control of a class of discrete event processes. *SIAM J. Control and Optimization, 25(1)*, pp. 206-230.

12] M. Shayman and R. Kumar, 1994. Supervisory control of nondeterministic systems with driven events via prioritized synchronization and trajectory models. *SIAM Journal of Control and Optimization, 33(2)*, pp. 469-497.

Forbidden State Avoidance in Controlled Petri Nets Under Partial Observation

L. Zhang and Lawrence E. Holloway*
Department of Electrical Engineering and
Center for Robotics and Manufacturing Systems
University of Kentucky
Lexington, Kentucky 40506

Abstract

Forbidden state avoidance control for controlled Petri nets has typically assumed full observation of the system state. However, in many situations, it is difficult if not impossible for a supervisor to observe uniquely all states or events. In such situations, some events (transition firings) may be completely unobservable, or two events may be indistinguishable to the supervisor. Under this partial observation, the supervisor's state estimate of the plant is a set of possible states corresponding to the observations. A control policy must be formulated that will ensure that the plant will not reach a forbidden state, no matter which state in the estimate set the plant is currently in. This paper represents an initial investigation of enforcement of state-based specifications using event based (partial observations. We first present a relatively compact method for representing the state estimate set. In particular, the state estimate set is represented as just the possible markings which are minimal nodes in the unobservable reachability space. We later show that this can be updated using an analysis of the path structure. Secondly, we present a general formulation of state control under partial event observation, and present a method of determining the control using a generalization of previous path algebra methods.

1 Motivation

Forbidden state avoidance control for controlled Petri nets has typically assumed full observation of the system state. However, in many situations, it is difficult if not impossible for a supervisor to observe uniquely all states or events. In such situations, some events (transition firings) may be completely unobservable, or two events may be indistinguishable to the supervisor. Under this partial observation, the supervisor's state estimate of the plant is a set of possible states corresponding to the observations. A control policy must be formulated that will ensure that the plant will not reach a forbidden state, no matter which state in the estimate set the plant is currently in.

In this paper, we consider the problem of forbidden state avoidance in controlled Petri nets under partial observation of events. Other researchers have considered event based control of

---

[*] Please address all correspondence to L. E. Holloway at the above address. This work has been supported in part by NSF grant ECS-9308737, Rockwell International, NASA grant NGT-40049, and the Center for Robotics and Manufacturing Systems at the University of Kentucky

Petri nets, but primarily in the context of event-based control specifications. Li and Wonham extended basic discrete event systems language based control results to the control of Petri nets, and showed that language based control can be achieved through a state-based supervisor [1]. Giua and Dicesare [2] and Kumar and Holloway [3] have also presented methods for control of language specifications using state based techniques. Sreenivas has shown that in general controllability of languages generated by PN's is undecidable [4].

In this paper, we consider enforcement of state-based specifications, rather than language specifications. However, the controller must make decisions based only on events observed. Furthermore, some transition firings may be unobservable, and others may be indistinguishable. This means that a state estimate of the set is a set of possible states, rather than just a single state. To achieve the state avoidance goals, it must be assured that all states within the current and future state estimate sets will be admissible.

In section 2, we overview controlled Petri nets. In section 3, we introduce reachability under partial observations, and show that the state estimate set can be represented by minimal nodes in the unobservable reachability space. Section 4 generalizes precedence path algebras for characterizing reachability. Section 5 then presents the maximally permissive state avoidance control, and shows that it can be evaluated using the reduced marking estimate set. Finally, we briefly discuss current research into recursively updating the reduced marking estimate set using path algebra results.

2 Controlled Partially Observed Petri Nets

A Controlled Partially Observed Petri Net is defined as $\mathcal{G} = (\mathcal{P},\mathcal{T},\mathcal{E},\mathcal{C},\mathcal{B},\Sigma,\mu)$,where $\mathcal{P}$ is a finite set of *state places* (sometimes referred to as just *places*), $\mathcal{T}$ is a finite set of *transitions*, $\mathcal{E} \subseteq (\mathcal{P}\times\mathcal{T})\cup(\mathcal{T}\times\mathcal{P})$ is a set of directed arcs connecting state places and transitions. Σ is a set of event labels, including the hidden event ϵ, and $\mu : \mathcal{T} \rightarrow \Sigma$ is an observation mask function, associating an event label to each transition. We define the set T_c of controlled transitions as those with a control place as an input, and the set T_o of observed transitions to include any $t \in \mathcal{T}$ with $\mu(t) \neq \epsilon$. T_{uo}, the set of unobserved transitions, is defined as $T_{uo} = \mathcal{T} - T_o$. In this paper we assume $T_c \subseteq T_o$.

Figure 1 illustrates a Controlled Partially Observed Petri Net where state places are indicated by circles,transitions by bars, and control places are boxes.

Given a transition t, the set of state places which are outputs to t is denoted $t^{(p)} \subseteq \mathcal{P}$, the set of state places which are inputs to t is denoted by ${}^{(p)}t \subseteq \mathcal{P}$, and the set of control places which are inputs to t is denoted by ${}^{(c)}t \subseteq \mathcal{C}$, In this paper, we assume that all transitions have a nonempty set of input state places [1], so ${}^{(p)}t \neq \emptyset$. Similarly, we let ${}^{(t)}p$ and $p^{(t)}$ indicate the transitions into and out of a state place p.

The *marking* (state) of a PN is a function $m : \mathcal{P} \longrightarrow \mathcal{N}$ indicating the number of *tokens* in each state place (where $\mathcal{N}$ is the set of nonnegative integers). Define the set of all markings for a Petri net $\mathcal{G}$ as $\mathcal{M} := \{m \mid \mathcal{P} \longrightarrow \mathcal{N}\}$. A marking is sometimes represented as *tokens* distributed over $\mathcal{P}$, where we say place p has n tokens if $m(p) = n$. Tokens are represented as black dots in figure 1. A control is a binary function $u : \mathcal{C} \longrightarrow \{0,1\}$ over control places, with the set of all controls denoted $\mathcal{U}$.

A transition is *state enabled* if each of its input places has at least one token. A set of transitions $T \subseteq \mathcal{T}$ is *state enabled* under a marking m when $m(p) \geq |p^{(t)} \cap T|$ for all $p \in {}^{(p)}t$.

---

[1] This assumption is made to simplify the theoretical development, but has no real impact on the class of systems that can be considered. A transition t with no input state places can be equivalently modeled by adding a marked place in a self loop with the transition t.

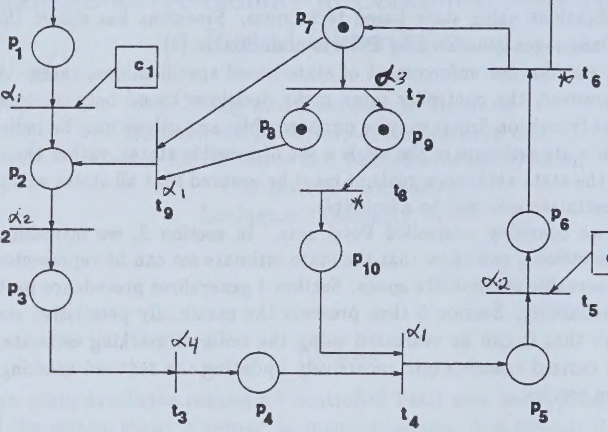


Figure 1: An example controlled Petri net.

The set of state enabled transitions is denoted $T_e(m)$. A set T of transitions is *control enabled* is $u(c) = 1$ for all $c \in {}^{(c)}t$ for all $t \in T$. A state change occurs when a set of enabled transitions $T \subseteq T_e(m)$ *fires*, which results in a new marking m' defined by the following equation:

$$m'(p) = m(p) - |p^{(t)} \cap T| + |{}^{(t)}p \cap T|$$

Thus, the marking of each state place is decremented by the firing of an output transition of the state place and incremented by the firing of an input transition to the state place.

2.1 Reachability and T-reachability

Given a marking m, a marking m' is said to be *immediately reachable* if there exists a set (possibly empty) of transitions $T \subseteq \mathcal{T}$ such that T is state enabled under m, and the firing of T results in the marking m'. The set of immediately reachable markings under a marking m is denoted $R_{T,1}(m)$, where T represents the set of transitions allowed to fire. A marking m_k is said to be *reachable* from a marking m_0 if there exists a sequence $(m_0 m_1 \ldots m_k)$ such that $m_i \in R_{T,1}(m_{i-1})$ for $0 < i \leq k$. The set of all reachable markings from m is denoted $R_{T,\infty}(m)$, with the convention that $m \in R_{T,\infty}(m)$, corresponding to the firing of a zero-length sequence of transition sets. Extending this notation, we note that $R_{T_{uo},1}(m,u)$. is the set of immediate markings reachable from m under u by concurrent firing of a set of unobserved transitions, and $R_{T_{uo},\infty}(m,u)$ is the set of markings reachable under any number of sets of unobservable transition firings.

Let F be a set of places representing a set of markings, $M_F := \{m \mid m(p) \geq 1, p \in F\}$. Furthermore, let the collection of sets $\mathcal{F} \subseteq 2^{\mathcal{P}}$ represent the set of markings $M_{\mathcal{F}} := \{m \in M_F \mid F \in \mathcal{F}\}$. Given a $\mathcal{F} \subseteq 2^{\mathcal{P}}$ and a set of transitions T, the set of *T-reachable* markings is defined as

$$N_T(M_{\mathcal{F}}) = \{m \mid R_{\mathcal{T}-T,\infty}(m) \cap M_{\mathcal{F}} \neq \emptyset\}$$

This is the set of markings from which some markings in $M_{\mathcal{F}}$ can be reached by transition firings not in T. If we let T be the set of controllable transitions, then $N_{T_c}(M_{\mathcal{F}})$ is the set of markings which can uncontrollably lead to a the set of markings specified by $\mathcal{F}$, and its complement, $\mathcal{M} - N_{T_c}(M_{\mathcal{F}})$ represents the admissible marking set, denoted $\mathcal{A}(\mathcal{M}_{\mathcal{F}})$ [5]. If we consider T as the set of observable transitions, then $N_{T_o}(M_{\mathcal{F}})$ is the set of markings which unobservably lead to $M_{\mathcal{F}}$.

The set of *T-reachable boundary markings* for a set of transitions T is defined as

$$B_T(M_{\mathcal{F}}) = \{m \notin N_T(M_{\mathcal{F}}) \mid R_{T,1}(m) \cap N_T(M_{\mathcal{F}}) \neq \emptyset\}$$

This is the set of markings which will lead to a T-reachable marking after firing some set of transitions in T. When T is the set of controllable transitions, then $B_{T_c}(M_{\mathcal{F}})$ is the set of markings which require immediate disabling of control transitions to prevent the uncontrollable reachability of markings in $M_{\mathcal{F}}$.

We want to develop predicates that can identify both sets $N_T(\mathcal{M}_{\mathcal{F}})$ and $B_T(\mathcal{M}_{\mathcal{F}})$.

3 Reachability under partial observation

We generalize the notion of observation masks to operate over transition sets, and then sequences of transition sets. Define the following:

$$\begin{aligned} \mu(T) &= \{\mu(t) \mid t \in T \cap T_o\} \\ \mu^{-1}(\gamma) &= \{T \subseteq T_o \mid \mu(T) = \gamma\} \quad \forall \gamma \subseteq \Sigma \end{aligned}$$

Note that $\mu(T) = \mu(T - T_{uo})$ for all $T \subseteq \mathcal{T}$, and $\mu(T) = \emptyset$ when $T \subseteq T_{uo}$. For all $\gamma \subseteq \Sigma$, $\mu^{-1}(\gamma)$ is the minimal set whose mask value is γ.

Finally, the notion of masks is generalized to sequences of transition sets $\mu : (2^{\mathcal{T}})^* \longrightarrow (2^{\Sigma})^*$, where $\forall \sigma_n = T_1 T_2 \ldots T_n$, $\mu(\sigma_n)$ is inductively defined as

$$n = 1 \quad \mu(\sigma_n) = \begin{cases} \varepsilon & T_1 \subseteq T_{uo} \\ \mu(T_1) & T_1 \not\subseteq T_{uo} \end{cases}$$

$$n > 1 \quad \mu(\sigma_n) = \begin{cases} \mu(\sigma_{n-1}) & T_n \subseteq T_{uo} \\ \mu(\sigma_{n-1})\mu(T_n) & T_n \not\subseteq T_{uo} \end{cases}$$

where ε is the empty string, and $\sigma_{n-1} = T_1 T_2 \ldots T_{n-1}$.

Consider that we are given an initial state m_0 and a sequence of observed symbol sets over time. Let α_i be the sequence up to the ith observed event set,that is $\alpha_i = \gamma_1\gamma_2\cdots\gamma_i$,where α_0 is defined as the empty string ε. The set of possible states after the ith observed event set will then be $\mathcal{M}(m_0, \alpha_i)$. Formally,

$$\mathcal{M}(m_0, \alpha_i) = \{m \mid m = \mathcal{R}(m_0, \sigma), \sigma \in \mu^{-1}(\alpha_i)\}$$

It is thus the largest set such that for every $m \in \mathcal{M}(m_0, \alpha_i)$,there exists a valid firing sequence σ from m_0 to m whose mask value is α_i.

We define $\overline{M}(m_0, \alpha_i)$ as the minimum cardinality set such that $M(m_0, \alpha_i) = \{m \in R_{T_{uo},\infty}(m') \mid m' \in \overline{M}(m_0, \alpha_i)\}$. Thus $M(m_0, \alpha_i)$ is the set of markings unobservably reachable from markings in $\overline{M}(m_0, \alpha_i)$. Note that $\overline{M}(m_0, \alpha_i)$ always exists, since $M(m_0, \alpha_i)$ is closed under unobserved transition firings. The lemma below follows directly from the above discussion, and states that T_o-reachability after an observed event sequence α can be determined by just considering markings in $\overline{M}(m_0, \alpha_i)$.

Lemma 1 Given any $\mathcal{F} \subseteq 2^{\mathcal{P}}$, event sequence α, and initial marking m_0, $M(m_0, \alpha) \cap N_{T_o}(\mathcal{M}_{\mathcal{F}}) \neq \emptyset$ only if $\overline{M}(m_0, \alpha) \cap N_{T_o}(\mathcal{M}_{\mathcal{F}}) \neq \emptyset$.

The lemma states that if a marking in $\mathcal{M}_{\mathcal{F}}$ is unobservably reachable from a marking in $M(m_0, \alpha)$, then it is unobservably reachable from a marking in the smaller set $\overline{M}(m_0, \alpha)$. This means that we will be able to determine a control by considering only the smaller set.

Finally, we present a lemma that will ensure that in determining control only one marking will needs to be considered. This will be true if all transitions which represent a conflict are observable transitions.

Lemma 2 Given a net $\mathcal{G}$, for every t such that $t \in p^{(t)}$ for some p with $|p^{(t)}| > 1$, if $t \in T_o$ and $|\mu^{-1}(\mu(t))| = 1$, then $|\overline{M}(m_0, \alpha)| = 1$ for any initial marking m_0 and any observable event sequence α that can occur from m_0.

Proof: The condition ensures that whenever there is a choice (conflict) as to which transition out of a marked place will fire, the choice will be observed by a unique event label. □

4 Precedence paths and reachability predicates

To characterize reachability, we define reachability predicates over a path algebra. In [6], reachability predicates over a path algebra were defined for uncontrolled reachability. In this section, we generalize the results of [6] to apply to unobservable reachability as well.

A path is a directed sequence of state places and transitions in the Petri net. This paper is exclusively concerned with the set of paths which begin and end with state places. Define $\Pi_{\mathcal{G}}$ as the set of all strings of alternating transitions and state places (which begin and end with state places) such that for any string $\pi \in \Pi_{\mathcal{G}}$, $p \in t^{(p)}$ for any substring tp in π, and $p \in {}^{(p)}t$ for any substring pt in π. For each path $\pi = (p_1 t_1 \ldots p_{n-1} t_{n-1} p_n) \in \Pi_{\mathcal{G}}$, the notation T_π indicates the set of transitions which are inputs to the path, i.e. $T_\pi = {}^{(t)}p_1$, and the notation p_π represents the terminating place in the path, i.e. $p_\pi = p_n$. When performing set operations, we will consider a path to be a subset of $\mathcal{T} \cup \mathcal{P}$. For example, $p \in \pi \cap \mathcal{P}$ means that p is a state place in the path π. We generalize the notion of markings to paths by defining $m(\pi) := \sum_{p \in \pi} m(p)$.

Given a net $\mathcal{G}$, define $J := \{p \in \mathcal{P} \mid |{}^{(t)}p| > 1\}$ as the set of *joining places.* Define $D \subseteq \mathcal{P}$ as any set of places such that $D \supseteq \{p \mid |p^{(t)}| > 1\} \cup \{p \mid \exists t \in p^{(t)} \text{ s.t. } t^{(p)} \cap J \neq \emptyset\}$. Thus, D can be defined as any set of places, as long as it contains all places with multiple output transitions, and all places which have output transitions which are inputs to joining places. For a set of places F and a set of transitions T, the set of *precedence paths* $\Pi_{F,T}(p) \subseteq \Pi_{\mathcal{G}}$ for $p \in D \cup F$ is defined below.

Definition 1 : Given a set of transitions $T \subseteq \mathcal{T}$,a set of places $F \subseteq \mathcal{P}$ and some place $p \in P_F$, define the set of *precedence paths* $\Pi_{F,T}(p) \subseteq \Pi_{\mathcal{G}}$ for p as the set of all paths $\pi = (p_1 t_1 \ldots t_{n-1} p_n)$ such that

1. Path ends at p: $p_n = p$;
2. Path contains no cycles: $p_i \neq p_j$ and $t_i \neq t_j$ for $1 \leq i \neq j \leq n$;
3. Path contains no transitions in T: $\pi \cap T = \emptyset$;
4. Restriction on branching, joining, and places in F: For $1 \leq i \leq n$, $p_i \in D \cup F$ implies $i = n$, and $p_i \in J$ implies $i = 1$.

5. Path starting condition for singleton transition input: If $\exists t \in \mathcal{T}$ such that ${}^{(t)}p_1 = \{t\}$, then $t \in T$ or ${}^{(p)}t \cap (D \cup F \cup \pi) \neq \emptyset$.

The set of precedence paths for all $p \in D \cup F$ for a given set of places F and a given set of transitions T is denoted as $\Pi_{F,T} := \cup_{p \in D \cup F} \Pi_{F,T}(p)$. The conditions 1-3 in the above definition are straightforward. The fourth condition indicates only the ending place of a path can be an element of $D \cup F$, and only the beginning place can be joining place. Statement 5 says that when the starting place is not a joining place but does have a transition input, then the transition input is either a transition in T, or has as input a place in $D \cup F$ or in the path itself.

We now define an algebra of expressions characterizing the net, and then present a method for evaluating these expressions. The expressions are defined using paths in $\Pi_{\mathcal{G}}$ as defined in the previous section, and using strings of transitions $\sigma \in \mathcal{T}^*$. The set of transition strings $\mathcal{T}^*$ includes the empty string ϵ.

Definition 2 : The set Expr of expressions in the algebra is defined recursively using the three binary operators: $+$, $\cdot$, and the *transition string prefix* operator.

1. $\pi \in \Pi_{\mathcal{G}} \Longrightarrow \pi \in$ Expr
2. $\texttt{expr}_1, \texttt{expr}_2 \in$ Expr $\Longrightarrow (\texttt{expr}_1 + \texttt{expr}_2) \in$ Expr, $(\texttt{expr}_1 \cdot \texttt{expr}_2) \in$ Expr
3. $\texttt{expr} \in$ Expr and $\sigma \in \mathcal{T}^* \Longrightarrow {}^{\sigma}(\texttt{expr}) \in$ Expr.

The operation between transition strings and expressions, given in item 3 above, is referred to as a transition string prefix operation.

An expression $\texttt{prod} \in$ Expr is in *product form* if it does not contain any "+" operations. Let the notation $\Pi(\texttt{expr}) \subseteq \Pi_{\mathcal{G}}$ represent the set of all paths occurring in expression $\texttt{expr}$. For a product form expression $\texttt{prod}$, define $\Sigma(\texttt{prod}, \pi) := \{\sigma \mid {}^{\sigma}\pi \text{ is a term in } \texttt{prod}\}$.

The expressions we consider in this paper are constructed by recursively performing an expansion operation, $\texttt{expand}_{F,T}(\texttt{expr})$ on an expression $\texttt{basic}_{F,T}(p)$. These are defined below.

Definition 3 : Given a set of transitions $T \subseteq \mathcal{T}$, and a set of places $F \subseteq \mathcal{P}$, define the following

1. For any place $p \in F \cup D$, with $\{\pi_1, \pi_2, \ldots \pi_n\} = \Pi_{F,T}(p)$, define

   $$\texttt{basic}_{F,T}(p) := \pi_1 \cdot \pi_2 \cdots \pi_n$$

2. For any $\pi \in \Pi_G$,

   $$\texttt{expand}_{F,T}(\pi) := \pi + \sum_{\substack{t \in T_\pi - T \\ and\ {}^{(p)}t \cap (D \cup F) \neq \emptyset}} {}^{t}\left(\prod_{p \in {}^{(p)}t \cap (D \cup F)} \texttt{basic}_{F,T}(p) \right)$$

3. For any expression $e \in$ Expr, $\texttt{expand}_{F,T}(e)$ is found by replacing in e (nonrecursively) each occurrence of $\pi \in \Pi_{F,T}(p)$ with $\texttt{expand}_{F,T}(\pi)$.
4. For any $p \in D \cup F$, $\texttt{expr}_{F,T}(p)$ is any expression such that : $\texttt{expr}_{F,T}(p)$ results from any number of recursive expansions of $\texttt{basic}_{F,T}(p)$, and $\texttt{expr}_{F,T}(p) \equiv \texttt{expand}_{F,T}(\texttt{expr}_{F,T}(p))$, i.e. is a fixed point to the expand operation under the equivalence relation defined in the next section.
5. For any $F = \{p_1, p_2, \ldots p_n\}$,the path expression for F is

   $$\texttt{expr}_T(F) := {}^{t_F}(\texttt{expr}_{F,T}(p_1) \cdot \texttt{expr}_{F,T}(p_2) \cdots \texttt{expr}_{F,T}(p_n))$$

   where t^F is an imaginary transition defined s.t. ${}^{(p)}t_F = F$.

6. for any $\mathcal{F} = \{F_1, F_2, \ldots F_k\} \subseteq 2^{\mathcal{P}}$,

$$\texttt{expr}_T(\mathcal{F}) := \texttt{expr}_T(F_1) + \texttt{expr}_T(F_2) + \cdots + \texttt{expr}_T(F_k)$$

Thus, $\texttt{basic}_{F,T}(p)$ is a product of paths ending with p, and the expand operation for a path π adds to π a sum of product terms, one for each input transition to the path which is not in T. The product term for each such transition t has the transition as a prefix string. The product then consists of the product of $\texttt{basic}_{F,T}(p)$ expressions for each place $p \in D \cup F$ which is an input to the transition. $\texttt{expr}_{F,T}(p)$ is a result of recursive expands on $\texttt{basic}_{F,T}(p)$ and $\texttt{expr}_T(F)$ is the product of these expressions for $p \in F$.

Two transitions $t_1, t_2 \in \mathcal{T}$, $t_1 \neq t_2$, are said to be *in conflict* if ${}^{(p)}t_1 \cap {}^{(p)}t_2 \neq \emptyset$. For the remainder of the paper, we assume the following condition.

Definition 4 : Given a PN G, a set of transitions T, and a set $\mathcal{F} \subseteq 2^{\mathcal{P}}$, let $T_{\mathcal{F}} = \{t \in T_\pi \cap T \mid \pi \in \Pi(\texttt{expr}(\mathcal{F}))\}$. $\mathcal{F}$ is said to satisfy the *precedence path input condition* (PPIC) if the following statements are true:

1. for any two paths $\pi_1, \pi_2 \in \Pi(\texttt{expr}_T(\mathcal{F}))$ such that $T_{\pi_1} \cap T_{\pi_2} \neq \emptyset$, if there exists some $t_1 \in p^{(t)}_{\pi_1}$ and $t_2 \in p^{(t)}_{\pi_2}$ such that t_1 and t_2 are in conflict, then $p_{\pi_1} = p_{\pi_2}$.
2. For any $t_1, t_2 \in T_{\mathcal{F}}$ with $t_1 \neq t_2$, ${}^{(p)}t_1 \cap {}^{(p)}t_2 = \emptyset$;
3. For any $t_1 \in T_{\mathcal{F}}$ and any $\pi \in \Pi(\texttt{expr}(\mathcal{F}))$, ${}^{(p)}t_1 \cap \pi = \emptyset$.

Condition 1 above, formerly referred to as the transition conflict condition in [6], states that if any two paths have a common input transition, and if any of their outgoing transitions are in conflict, then the paths must have the same terminating place. We note that the net classes of marked graphs and state graphs trivially satisfy the TC condition. Furthermore, note that most nets not satisfying condition TC can be easily modified such to satisfy the condition by breaking either of the paths into two subpaths.

Definition 5 : Given any two distinct sequences $\sigma_1, \sigma_2 \in T^*$, σ_1 and σ_2 are said to be *in conflict* if $\sigma_1 = \sigma_1' t_1 \sigma$ and $\sigma_2 = \sigma_2' t_2 \sigma$ for some common suffix σ, where t_1 and t_2 are in conflict. Given a product form expression $\texttt{prod}$ and some $\pi \in \Pi(\texttt{prod})$, the *token threshold* $Q(\texttt{prod}, \pi)$ is the maximum number of strings in $S(\texttt{prod}, \pi)$ that are jointly in conflict. Given a set $S \subseteq S(\texttt{prod}, \pi)$, if all distinct $\sigma_1, \sigma_2 \in S$ are in conflict, and $|S| = Q(\texttt{prod}, \pi)$, then S is referred to as a *maximum cardinality conflict set* for $S(\texttt{prod}, \pi)$.

Note from the above definition that if $S(\texttt{prod}, \pi) \neq \emptyset$ but there are no two σ_1 and σ_2 in $S(\texttt{prod}, \pi)$ that are in conflict, then the largest subset of conflicting strings in $S(\texttt{prod}, \pi)$ is a singleton set, so $Q(\texttt{prod}, \pi) = 1$. When $S(\texttt{prod}, \pi) = \emptyset$, then $Q(\texttt{prod}, \pi) = 0$.

A *marking valuation predicate* is a function that assigns a binary value to each $\texttt{expr} \in \textsc{Expr}$. Both predicates will depend on a common set of properties, which we refer to as the basic valuation predicate axioms.

Definition 6 : Given any predicate $V : \texttt{expr} \to \{0, 1\}$, the predicate is said to satisfy the *basic valuation predicate axioms* if the following properties hold for any expressions $e_1, e_2, e_3 \in \textsc{Expr}$:

1. Sum is Disjunctive:	$V_m(e_1 + e_2) = 1$ iff $V_m(e_1) = 1$ or $V_m(e_2) = 1$
2. Distributive $\cdot$ over $+$:	$V_m(e_1(e_2 + e_3)) = V_m(e_1e_2 + e_1e_3)$
3. Distributive prefix over $+$:	$V_m({}^{\sigma}(e_2 + e_3)) = V_m({}^{\sigma}e_2 + {}^{\sigma}e_3)$
4. Distributive prefix over $\cdot$:	$V_m({}^{\sigma}(e_1e_2)) = V_m({}^{\sigma}(e_1)\,{}^{\sigma}(e_2))$
5. Concatenation through prefix:	If $\sigma = \sigma_1\sigma_2$, then $V_m({}^{\sigma_1}({}^{\sigma_2}e_1)) = V_m({}^{\sigma}e_1)$

We can now define the predicates we will use in this paper.

Definition 7 : Given a marking m and a transition set T, the *T-reachable predicate* for m is the binary valuation $\Lambda_m : \textsc{Expr} \longrightarrow \{0,1\}$ that satisfies the following properties:

1. Λ_m satisfies the basic valuation predicate axioms, and
2. For any product $\texttt{prod}$ in $\textsc{Expr}$, $\Lambda_m(\texttt{prod}) = 1$ iff for all $\pi \in \Pi(\texttt{prod})$:

$$m(\pi) \geq Q(\texttt{prod}, \pi)$$

Definition 8 : Given a marking m and transition set T, the *boundary T-reachable predicate* for m is the binary valuation $\Delta_{m,T} : \textsc{Expr} \longrightarrow \{0,1\}$ that satisfies the following properties:

1. $\Delta_{m,T}$ satisfies the basic valuation predicate axioms.
2. For any product $\texttt{prod}$ in $\textsc{Expr}$, $\Delta_m(\texttt{prod}) = 1$ iff for all $\pi \in \Pi(\texttt{prod})$,

$$m(\pi) \geq Q(\texttt{prod}, \pi) - |T_\pi \cap T_e(m) \cap T|$$

Comparing Δ_m to Λ_m, we note that for Δ_m, the token thresholds for the paths in the product are reduced by the number of enabled transitions in T which are inputs to the path. Note also that in the above definition for any $\texttt{expr} \in \textsc{Expr}$ and any $m \in \mathcal{M}$, $\Lambda_m(\texttt{expr}) = 1$ implies $\Delta_m(\texttt{expr}) = 1$. We define the equivalence relationship $\equiv$ such that for any $\texttt{expr}_1, \texttt{expr}_2$, $\texttt{expr}_1 \equiv \texttt{expr}_2$ if for all markings m, $\Lambda_m(\texttt{expr}_1) \equiv \Lambda_m(\texttt{expr}_2)$ [6]. Note that the expression $\texttt{expr}_F(p)$ for a place p (definition 3) was defined as a fixed point under this equivalence relation.

We finally have the following two theorems, both of which are direct generalizations of the results of [6].

Theorem 1 Given a PN $\mathcal{G}$,a set of transitions T and a $\mathcal{F}$ satisfying the TC condition, for any marking $m \in \mathcal{M}$, $m \in N_T(M_\mathcal{F})$ if and only if $\Lambda_m(\texttt{expr}_T(\mathcal{F})) = 1$.

Theorem 2 Given a PN $\mathcal{G}$,a set of transitions T and a $\mathcal{F}$ satisfying the PPIC, for any marking $m \in \mathcal{M}$, $m \in B_T(M_F)$ if and only if $\Delta_{m,T}(\texttt{expr}_T(\mathcal{F})) = 1$ and $\Lambda_m(\texttt{expr}_T(\mathcal{F})) = 0$.

5 Control Synthesis under partial observation

Consider a control goal to avoid a set $\mathcal{M}_\mathcal{F}$ of markings specified by $\mathcal{F} \subseteq 2^\mathcal{P}$. Since $N_{T_c}(\mathcal{M}_\mathcal{F})$ is the set of markings which can uncontrollably lead to markings in $\mathcal{M}_\mathcal{F}$, then define $\mathcal{A}(\mathcal{M}_\mathcal{F}$ as the set of *admissible markings*, i.e. markings for which a control exists to prevent reaching markings in $\mathcal{M}_\mathcal{F}$.

An *event-based control policy* is a function $U : \mathcal{M} \times \Sigma^* \longrightarrow 2^\mathcal{U}$, where $U(m_0, \alpha)$ maps an initial marking m_0 and a sequence of observed events onto a set of controls. Note that the policy can be nondeterministic, since it may return several controls for any given m_0, α.

We desire a control policy that will ensure that all markings up to and immediately following the next observed event must lie within the admissible markings set. Since we would like a control policy that provides all allowable control options, then using the property that $M' \subseteq R_{T,1}(M')$ for any M' and T, we get the following definition:

Definition 9 : An event-based control policy U_{m_0} is *maximally permissive* with respect to forbidden set of markings $M_\mathcal{F}$ if for any marking m_0 and observed event string α,

1. $u \in U(m_0, \alpha)$ and $M(m_0, \alpha) \subseteq \mathcal{A}(\mathcal{M}_\mathcal{F})$, then $R_{T_o,1}(R_{T_{uo},\infty}(M(m_0, \alpha), u), u) \subseteq \mathcal{A}(\mathcal{M}_\mathcal{F})$, and
2. for any $u \notin U(m_0, \alpha)$ and $M(m_0, \alpha) \subseteq \mathcal{A}(\mathcal{M}_\mathcal{F})$, then $R_{T_o,1}(R_{T_{uo},\infty}(M(m_0, \alpha), u), u) \not\subseteq \mathcal{A}(\mathcal{M}_\mathcal{F})$

Note that the above maximally permissive policy defined above is nondeterministic. It is also non-unique since there is no specification of what a suitable control is when the current marking estimate set $M(m_0, \alpha)$ is not admissible. However, it is evident that if $m_0 \in \mathcal{A}(\mathcal{M}_\mathcal{F}))$, then all reachable markings under the control policy will also be in the admissible marking set.

Define for a marking m, a control u, and a path π:

$$C(m, u, \pi) := |t \in T_\pi \cap T_e(m) \cap T_c \mid u(c) = 1 \forall c \in {}^{(c)}t\}|$$

$C(m, u, \pi)$ is the number of enabled controlled transitions which are inputs to the path π. This leads us to the following theorem.

Theorem 3 For a given set $\mathcal{F}$ of forbidden conditions satisfying the $PPIC$, the maximally permissive event-based control policy for an initial marking m_0 and event sequence α is the set of u such that for each $m \in \overline{M}(m_0, \alpha)$ and for each product *prod* in the sum-of-product form of *expr*$(\mathcal{F})$,there exists some $\pi \in \Pi(prod)$ for which $\mathcal{C}(m, u, \pi) < \mathcal{Q}(prod, \pi) - m(\pi)$.

Proof: From [6], the theorem statement will apply if we consider all $m \in M(m_0, \alpha)$. Recall that $\overline{M}(m_0, \alpha)$ was defined such that $M(m_0, \alpha) = \{m \in R_{T_{uo},\infty}(m') \mid m' \in \overline{M}(m_0, \alpha)\}$. Since $M(m_0, \alpha)$ is by its definition closed under unobservable transition firings, and since $T_c \subseteq T_o$, then $R_{T_{uo},\infty}(M(m_0, \alpha), u) = R_{T_{uo},\infty}(\overline{M}(m_0, \alpha), u)$ for any u. Thus, the theorem statement also applies for just considering the set $\overline{M}(m_0, \alpha)$. □

6 Discussion

We have presented an approach for enforcement of state-based specifications under partial event-based observations. This uses a reduced state estimate set for control calculation, and it is shown that this is equivalent to using a complete state estimate set.

One issue not explored in this paper is the update of the reduced state estimate set $\overline{M}(m_0, \alpha)$. Current research is developing efficient methods for updating this set each time a set of events is observed. The update can be accomplished in two steps. First, we note that the occurrence of the event set indicates that places that are inputs to the appropriate transitions must be marked, reachable from a current marking estimate. The enabling of these transitions can be represented as a condition set $F \subseteq \mathcal{P}$, and so the reachability of this enabling marking can be evaluated using reachability predicates over the unobservably reachable space. Then, for each marking in the prior set estimate, if there exists a product term in the expression for F that evaluates to one under the marking, then that marking could have led to the heard event set. The product term that evaluates to one also contains information (in the transition string prefix) of how the marking must have progressed to lead to the firing. Thus, it also tells how to update the marking estimate set.

References

[1] Y. Li and W.M. Wonham. Control of vector discrete-event systems I – the base model. *IEEE Transactions on Automatic Control*, 38(8):1214–1227, August 1993.

[2] Alessandro Giua and Frank DiCesare. Supervisory design using Petri nets. In *Proceedings of the 30th IEEE Conference on Decision and Control*, Brighton, UK, December 1991.

[3] Ratnesh Kumar and Lawrence Holloway. Supervisory control of deterministic Petri net languages with regular specification languages. *IEEE Transactions on Automatic Control*, February 1996. to appear.

[4] R.S. Sreenivas. On a weaker notion of controllability of a language K with respect to a language L. *IEEE Trans. on Automatic Control*, 38(9):1446–1447, September 1993.

[5] L. E. Holloway and B. H. Krogh. Controlled Petri nets: a tutorial survey. In *Proceedings of 11th International Conference on Analysis and Optimization of Systems: Discrete Event Systems*, Sophia-Antipolis, France, June 1994.

[6] L.E. Holloway and X. Guan. A generalization of state avoidance policies for controlled Petri nets. Technical Report CRMS-TR-93-2, Center for Robotics and Manufacturing Systems, University of Kentucky, Lexington, Kentucky, 1993. submitted to IEEE Transactions on Automatic Control.

A Discrete-Event Systems Approach to Modeling Dextrous Manipulation

S. L. Ricker** N. Sarkar*†* K. Rudie‡**
*Department of Computing and Information Science
†Department of Mechanical Engineering
‡Department of Electrical and Computer Engineering
Queen's University, Kingston, Ontario, Canada K7L 3N6

Abstract

To perform dextrous manipulation efficiently, it is necessary to coordinate the interactions of many component processes. This paper investigates one approach to coordination: discrete-event systems. The applicability of discrete-event systems to the modeling of dextrous manipulation tasks is studied. Discrete-event control theory offers formal methods for determining whether a coordinator of the components can be generated. The planar Grasp-Lift-Replace task of Howe and Cutkosky is presented as a discrete-event process. The task is modified to accommodate the inclusion of two-fingered exploratory procedures. The effectiveness of our discrete-event system is illustrated by examples of representative test cases.

1 Introduction

In many robotics problems, robot hands must perform delicate and precise operations that include grasping and smoothly lifting an object. Robotics research has addressed many components of this dextrous manipulation problem. These issues include multi-arm or finger coordination for manipulating objects [11], optimally distributing the load among different arms [18], decomposing the grasp force into equilibrating and interacting force fields [9], and allowing contact motion while manipulating an object [1]. While efficient dextrous manipulation requires modeling a controller for each component process (e.g., controlling contact force), it is also necessary to model a mechanism by which the component processes can be enabled and disabled in a preferred sequence.

Dextrous manipulation is described as "a process characterized by tactile events and control discontinuities" [2]. In this context, events mark transitions from one distinct phase of the task to the next. The need for techniques to facilitate a smooth progression through the discrete phases of the manipulation task is presented in [2]. Our work represents an initial effort to use a theoretical framework for the coordination of these

---

*The first two authors are partially funded by R. Ellis through grants from the Information Technology Research Centre, the Manufacturing Research Corporation of Ontario, and the federal government under the Institute for Robotics and Intelligent Systems (which is a national Centres of Excellence program). The last author is partially supported by the Natural Sciences and Engineering Research Council of Canada.

"control discontinuities". We apply the discrete-event control theory of Ramadge and Wonham [12] to the high-level description and coordination of a dextrous manipulation task. This leads to the design of a controller that permits only a set of desired actions to occur, while inhibiting undesirable behavior. The discrete-event systems approach provides a systematic procedure for generating strategies that are guaranteed to achieve a feasible task goal.

Discrete-event control theory has been used to model a variety of robotic applications including the coordination of mobile robotic agents [8], and a grasping task performed under the supervision of a vision system [15]. Our approach differs in that we present a working example of both the supervisor and the plant using the control-theoretic approach of Ramadge and Wonham, thereby performing supervisor synthesis. There are different ways to represent a discrete-event system e.g., Petri nets, finite state machines, and formal logic. We use finite state machines to model a dextrous manipulation task because the finite state machine formalism is sufficiently expressive for our purposes (i.e., all languages of interest in this paper are regular languages). Additionally, we would like to exploit the results associated with the body of work that already exists in discrete-event control theory and that uses finite state machines.

First, we present a description of the task and state the assumptions required for the discrete-event systems model. We discuss possible strategies for using two fingers to perceive several object properties during the manipulation task. Next, a brief background of discrete-event control theory is provided. Finally, the model is presented and some representative simulation results are discussed.

2 Background

The purpose of this paper is to investigate a discrete-event systems approach to designing a high-level supervisory controller for a dextrous manipulation task. We select a task (described in Section 2.1) that has received considerable attention in the study of both biological and robotic tactile systems and has not been modeled by a discrete-event system formalism. In addition, we propose a modeling strategy that allows exploratory procedures (EPs) to be carried out in the course of the task. There are some tasks, like the Grasp-Lift-Replace task, where it is feasible to determine object properties incorporated as part of the manipulation task. To that end we indicate how two existing exploratory procedures, a weight EP and a stiffness EP, could occur with minimal intrusion during the Grasp-Lift-Replace task.

2.1 The Grasp-Lift-Replace Task

Johansson and Westling [7] addressed the role of tactile signals during manipulations performed by — in particular, manipulating objects using a precision grip. The task selected for their series of studies is deceptively simple: an object is grasped between the thumb and index finger, lifted vertically off the table, and subsequently replaced on the table. The grasp-lift-replace task is divided into a series of distinct phases where the transition to the next phase is triggered by one or more of the four types of tactile units in the fingertips. To perform the manipulation task smoothly, it is necessary to sense the coefficient of friction at the area of contact.

The series of experiments conducted by Johansson and Westling has contributed much towards an understanding of how tactile sensors in the hands control fine manipulation

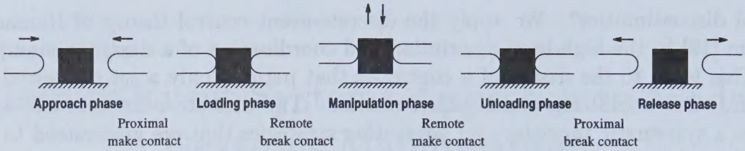


Figure 1: Phases of a Grasp-Lift-Replace task and the tactile events that distinguish phases as adapted from studies by Johansson and Westling(1987) by Howe *et al* (1990); after Howe *et al* (1990).

tasks. To determine whether similar mechanisms would aid in the control of robot manipulation tasks, Howe and Cutkosky [6] apply the hypotheses of the human studies to a robotic system. The robotic Grasp-Lift-Replace task involves five phases which are linked by four contact events (see Figure 1): the **approach** phase - the two fingers move towards the object in anticipation of the grasp; the **loading** phase - when both fingers make contact with the object, the grip (horizontal) force is increased simultaneously with the load (vertical) force until the object is both secure in the grasp and lifted off the table, signaling the beginning of the next phase; the **manipulation** phase - contact between the object and the table is broken, the object is lifted to a pre-determined height and then returned to the table ; the **unloading** phase - the object is on the table and the grip force is decreased; and the **release** phase - the fingers break contact with the object and move away to a neutral position. A change in contact marks the transition from one phase to another. In fact, the contact events also indicate a change from one low-level control mode to another. For instance, the end of the approach phase is indicated by contact between the fingers and the block (proximal make contact) and signifies a change from position to force control.

Robotic tactile sensors, described in [5],[16], detect the contact events and trigger the transition through the phases of the Grasp-Lift-Replace task. The specialized sensors detect slip during finger/object contact, as well as vibrational information to identify the remote contact events.

2.2 Exploratory Procedures

Exploratory procedures (EPs), first described by Lederman and Klatzky [10], elucidate a set of hand configurations and actions that humans consistently use to determine specific properties of objects. The structured nature of the EPs provides a framework for the development of similar procedures for robotic hands. Robotic EPs have been implemented for dextrous manipulators [17] and for robot fingers [3]. Some of the sampled object properties include: gross object size (volume), surface texture, thermal properties, weight, gross object shape, hardness/softness, and the shape of the contact area.

3 Task Description

We have selected the Grasp-Lift-Replace manipulation task of Howe and Cutkosky [6], with an additional step to determine stiffness, to be modeled as a discrete-event system. This provides a simple and concrete example against which we test the modeling framework. Information available from the manipulation task includes three important properties: coefficient of friction of contact, weight and stiffness of the object. These properties play a significant role in grasping and manipulation and sometimes need to be determined through tactile exploration.

To conform to the experimental set-up of Howe and Cutkosky, the following restrictions apply:

- The objects are block-shaped.
- The location of the object in space is known.
- Each finger is equipped with a two-axis force sensor, to measure the grasp and load forces, and a multi-element stress-rate sensor [16], to detect incipient slip.
- It is assumed that the dynamic tactile sensors are able to distinguish incipient slip from other vibrations.
- Fingertip position information is available.
- Once the coefficient of friction is estimated, the subsequent adjustment to the grasp force is sufficient to ensure no further slip occurs.

To impose some reasonable constraints on the discrete-event system, the following assumption is made: there is a maximum number of attempts allowed for aligning the fingers on either side of the object and for achieving stable contact — exceeding this threshold results in starting the task again or, in the latter case, failure of the task with no re-initiation permitted.

The task, as a discrete-event system, retains all the same phases as noted in Figure 1; however, the Stiffness EP is conducted during the manipulation phase. After successful completion of the approach phase, the Friction EP occurs. The coefficient of friction is calculated from force signals immediately preceding the detection of incipient slip [16]. Starting with a small grip force (loading phase), the load force is applied to lift the object (manipulation phase). When slip is detected (by the multi-element stress rate sensor), the grip force is increased, and another attempt is made to lift the object. This procedure repeats until the maximum number of increments is exceeded or the object is successfully lifted. The lifted object is held in static equilibrium at the end of the first manipulation phase. From this EP we measure the minimum grip force, F_g, and the load force, F_l.

The weight of the object, W, can be calculated with an estimate of the coefficient of friction, μ, for the contact between the finger and the block, and the grip force : $W = 2\mu F_g = F_l$.

The Stiffness EP is patterned after the one finger "Hardness" EP of Dario *et al.* [3] and constitutes the squeezing phase of the task. With the object held stationary at the end of the first manipulation phase, the gripping force is increased by ΔF_g which we call the squeezing force. This produces a deformation of the object by an amount Δx, which can be determined by the position of the fingertips. The stiffness is the ratio of the squeezing force to the deformation produced, $K = \Delta F_g / \Delta x$. It is possible that the deformation produced at each finger is different, though the squeezing force is the same, because the object is not homogeneous. In that case, the stiffness is taken as the average of the two.

4 Modeling the Dextrous Manipulation Task

The *plant* under consideration is the two-fingered manipulator, as described in [6], performing a Grasp-Lift-Replace (GLR) task. Our goal is to find a *supervisor* to monitor

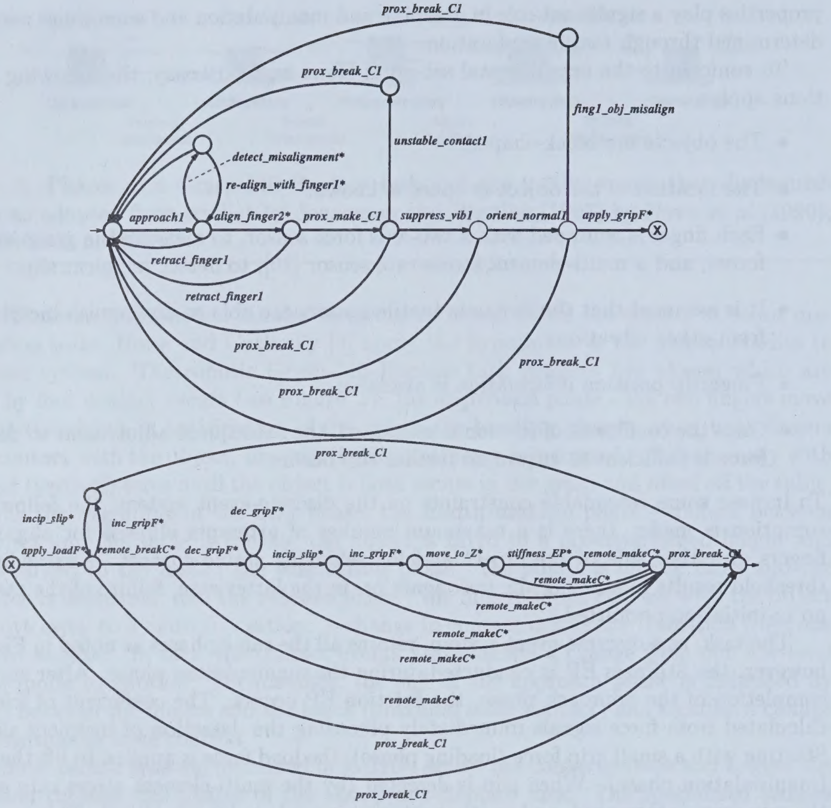

prox_break_C1
prox_break_C1
fing1_obj_misalign
unstable_contact1
detect_misalignment*
re-align_with_finger1*
approach1
align_finger2*
prox_make_C1
suppress_vib1
orient_normal1
apply_gripF*
X
retract_finger1
retract_finger1
prox_break_C1
prox_break_C1
prox_break_C1
prox_break_C1
incip_slip*
inc_gripF*
dec_gripF*
X
apply_loadF*
remote_breakC*
dec_gripF*
incip_slip*
inc_gripF*
move_to_Z*
stiffness_EP*
remote_makeC*
prox_break_C1
remote_makeC*
remote_makeC*
remote_makeC*
remote_makeC*
remote_makeC*
prox_break_C1
prox_break_C1

Figure 2: Finite-state automaton of Finger 1 for the Grasp-Lift-Replace task. Due to space limitations, the automaton has been separated into two parts, where state X of the top diagram is understood to be state X of the bottom diagram.

---

the behaviour of the plant. A supervisor enables or disables the *controllable* events of the plant in a manner that allows as much behaviour as possible subject to a prescribed set of constraints (i.e., the legal language E) and the set of uncontrollable events. Both the plant and the supervisor are modeled as finite-state machines.

The dextrous manipulator has two fingers, and therefore we model the actions of each finger separately. However, there are some aspects of the GLR task (e.g., remote make contact) in which the behaviour of both fingers must be synchronized. To accommodate these situations, our plant, G, is the the synchronous product of the finite-state automata for Finger 1 and Finger 2: $G{=}FING1 \| FING2$. An event occurs in the synchronous product only if it occurs in all automata in which the event appears. Shared events lead to coupled transitions which model the actions where the fingers operate concurrently. For a more formal definition, see [4]. The resulting automaton, for our dextrous manipulator, which defines all possible coordinated and independent behavior of the two fingers, has a total of 90 states and 242 transitions.

There are many ways in which the GLR task can be decomposed into a set of discrete events. We chose to describe each phase of the GLR task as a sequence of subtasks. While these subtasks are not necessarily comprehensive, the set of events we describe for our discrete-event system captures sufficient detail to illustrate the applicability of our approach. For example, planning a finger trajectory to the object is not specifically enumerated and is subsumed by the *approach*$_1$ and *approach*$_2$ events. Similarly, there is an assumption that underlying control exists to coordinate the continuous-time dynamics of the system. The set of events is (where $i = 1,2$ for Fingers 1 and 2 respectively)

- *approach*$_i$ is the approach of Finger 1 or 2 towards the object;
- *align_finger2** is the pre-contact alignment of Fingers 1 and 2 : Finger 1 is used as a point of reference to determine a contact location for Finger 2;
- *prox_make_C*$_i$ indicates that contact is made between Finger i and the object;
- *suppress_vib*$_i$ means it is necessary to suppress any contact vibration between the finger and the object;
- *orient_normal*$_i$ is the action of rotating the finger at the contact location until the finger is oriented normally to the object;
- *apply_gripF** is the application of a horizontal force to the object to secure it between the fingers;
- *apply_loadF** is the application of a vertical force to overcome gravity;
- *dec_gripF** is the incremental decrease of grip force to initiate incipient slip for purposes of determining the coefficient of friction at the contact point;
- *move_to_Z** is the event requiring the object to be lifted to a pre-specified height above the table;
- *remote_breakC** is the tactile sensor signal that the object has broken contact with the table;
- *stiffness_EP** indicates that the Stiffness EP occurs;
- *remote_makeC** is the event signaled by a tactile sensor which indicates that the object is once again in contact with the table;
- *prox_breakC*$_i$ is the retraction of the fingers and signals the end of contact between the fingers and the object;
- *detect_misalignment** occurs when Finger 2 is not lined up in space with Finger 1: to check for this error state we assume position information available to determine that the grip force is approximately collinear;
- *re-align_with_finger1** occurs when the two fingers are not lined up in space and Finger 2 must be moved to a position opposite to the reference point Finger 1;
- *unstable_contact*$_i$ is the result of an inability to suppress contact vibrations and local contact is not sufficiently stable to continue the task;
- *retract_finger*$_i$ is the removal of each finger from the proposed contact location — unlike *prox_breakC*, no contact has been made and fingers are simply withdrawn;

- *fing_obj_misalign$_i$* indicates that a finger cannot be oriented normally to the object;
- *inc_gripF** is an incremental increase in the grip force in response to the tactile sensing of slip at a contact point;
- *incip_slip** is the tactile sensing of incipient slip at a contact point.

The uncontrollable set of events for the two fingers are

$$\begin{aligned}\Sigma_{uc} &= \{detect\_misalignment^*, unstable\_contact_1, unstable\_contact_2, \\ & \quad fing_1\_obj\_misalign, fing_2\_obj\_misalign, incip\_slip^*.\}\end{aligned}$$

The plant has only two marker (terminal) states: the initial state prior to the start of the task, and the final state resulting from both fingers breaking contact with the object at the conclusion of the task. Note that because the initial state is marked, the task of "doing nothing" is considered a completed task.

To construct a supervisor for G, we first detail the desired or legal behavior of G. We present four specifications, each of which can be expressed as a finite-state automaton, that define the legal language E:

1. Finger 1 approaches the object before Finger 2.
2. Only two consecutive attempts (per approach) are allowed for aligning the two fingers prior to contact.
3. A total of two attempts to make contact will be permitted. After the third unsuccessful attempt, the task terminates with both fingers returning to a starting position.
4. If incipient slip is detected just after the load force is applied, only two attempts to adjust the grip force are allowed. When slip is sensed a third time, the task terminates and both fingers return to a starting position.

The automaton for each specification is shown in Figure 3. We calculate E by taking the intersection of the four specifications: $E = \text{Spec}_1 \cap \text{Spec}_2 \cap \text{Spec}_3 \cap \text{Spec}_4$. The legal language has 1412 states and 13724 transitions. Again, it is important to observe that the legal language presented here is one of many possible combinations of constraints that could be prescribed for the GLR task. The control problem we are interested in requires that the supervisor $\mathcal{S}$ must exactly impose the legal behaviour on G. Thus we need an automaton that recognizes only the desirable sequences as described by E. In addition we need a set of control rules that will indicate whether or not a given event at the current state of the plant is enabled or disabled. A supervisor that will solve our problem can be constructed as follows: first, we compute the largest controllable sublanguage of E; then we define a supervisor that ensures that only those strings and all those strings of G that are in this controllable sublanguage are permitted to occur. A supervisor does exist for the GLR task and has 1318 states and 2684 transitions.

Software developed in the Systems Control Group in the Department of Electrical Engineering at the University of Toronto (TCT931124), under the supervision of W. M. Wonham, was used to compute all results reported here. This software tool performs operations on discrete-event systems such as finding the supremal controllable sublanguage of a given legal language. We present representative results of our supervised discrete-event system $\mathcal{S}/G$ in the next section.

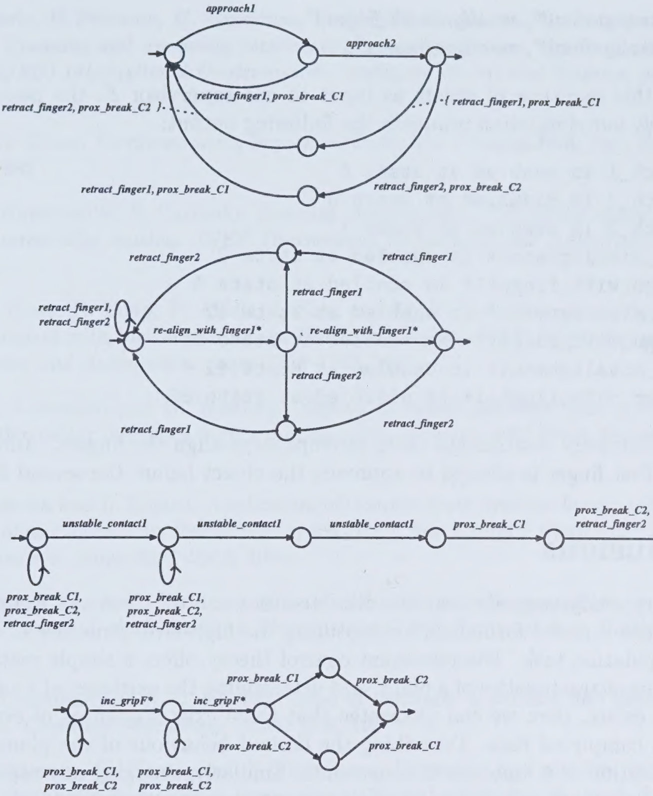

approach1
approach2
retract_finger1, prox_break_C1
retract_finger2, prox_break_C2
retract_finger1, prox_break_C1
retract_finger1, prox_break_C1
retract_finger2, prox_break_C2
retract_finger2
retract_finger1
retract_finger1
retract_finger1,
retract_finger2
re-align_with_finger1*
re-align_with_finger1*
retract_finger2
retract_finger1
retract_finger2
unstable_contact1
unstable_contact1
unstable_contact1
prox_break_C1
prox_break_C2,
retract_finger2
prox_break_C1,
prox_break_C2,
retract_finger2
prox_break_C1,
prox_break_C2,
retract_finger2
inc_gripF*
inc_gripF*
prox_break_C1
prox_break_C2
prox_break_C2
prox_break_C1
prox_break_C1,
prox_break_C2
prox_break_C1,
prox_break_C2

Figure 3: Specifications 1, 2, 3 and 4 expressed as finite-state automata. It is assumed that all other events in Σ are self-looped (i.e., can occur at every state).

5 Simulation Results

We have conducted a series of simulations to determine that the supervisor takes appropriate actions for any legal sequences of plant behavior. The simulation takes a desired sequence of events as input, performs a simple breadth-first search of the plant and the supervisor, and, using the feedback map, ascertains the status of each event (disable, enable). To illustrate that our controller correctly disables events, we selected the case where the fingers were not aligned on either either side of the block after three consecutive attempts to re-align the fingers.

5.1 Problems aligning the fingers

This sequence tests specification two: the task is successfully started, but more than two consecutive efforts are attempted to align the fingers in space.

> $approach_1$, $approach_2$,
> *detect_misalignment**, *re-align_with_finger1**,

*detect_misalignment**, *re-align_with_finger1**,
*detect_misalignment**, *re-align_finger1**.

When we use this sequence of events as input to our supervisor $\mathcal{S}$, the plant G, and feedback map ψ, our simulation produces the following output:

```
Event approach_1 is enabled at state 0
Event approach_2 is disabled at state 0
Event approach_2 is enabled at state 1
Event detect_misalignment* is enabled at state 2
Event re-align_with_finger1* is enabled at state 6
Event detect_misalignment* is enabled at state 12
Event re-align_with_finger1* is enabled at state 24
Event detect_misalignment* is enabled at state 42
Event re-align_with_finger1* is disabled at state 69
```

The controller correctly disables the third attempt to re-align the fingers. Additionally, at state 0, the first finger is allowed to approach the object before the second finger.

6 Conclusions

This preliminary study suggests that the discrete-event control theory of Ramadge and Wonham provides a useful formalism for capturing the high-level structure of a robotic dextrous manipulation task. Discrete-event control theory offers a simple methodology for describing the characteristics of a plant, and determining the existence of a supervisor. If a supervisor exists, then we can guarantee that there exist sequences of events that will result in a completed task. Describing the desired behaviour of the plant requires only the specification of a finite set of constraints. Similarly, changing any aspect of the plant's "good" behaviour is a matter of adding or removing a constraint.

We have considered the centralized control problem where one supervisor tracks the behavior of the plant, and all events are observable. Our study revealed that more complicated models capture the richness of a representative sensory-driven dextrous manipulation task. Future work would include an examination of timed or even hybrid models to address the issue of providing a more accurate description of sensor behaviour.

Acknowledgements

The authors would also like to thank R. E. Ellis and P. E. G. Gogolek for their helpful comments on the manuscript, and W. M. Wonham for providing the TCT software.

References

[1] A. Cole, J. Hauser, and S. Sastry. Kinematics and control of multifingered hands with rolling contact. In *Proceedings of the 1988 International Conference on Robotics and Automation*, pages 228–233, Philadelphia, Pennsylvania, April 1988.

[2] M. R. Cutkosky and J. M. Hyde. Manipulation control with dynamic tactile sensing. In *6th ISRR, Hidden Valley, Pennsylvania*, 1993.

[3] P. Dario, P. Ferrante, G. Giacalone, L. Livaldi, B. Allotta, G. Buttazzo, and A. Sabatini. Planning and executing tactile exploratory procedures. In *Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1896–1903, 1992.

[4] C.A.R. Hoare. *Communicating Sequential Processes.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1985.

[5] R.D. Howe and M. R. Cutkosky. Dynamic tactile sensing: Perception of fine surface features with stress rate sensing. *IEEE Transactions on Robotics and Automation*, 9(2):140–151, 1993.

[6] R.D. Howe, N. Popp, P. Akella, I. Kao, and M. R. Cutkosky. Grasping, manipulation, and control with tactile sensing. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1258–1263, 1990.

[7] R. S. Johansson and G. Westling. Signals in tactile afferents from the fingers eliciting adaptive motor responses during precision grip. *Experimental Brain Research*, 66:141–154, 1987.

[8] J. Košecká and L. Bogoni. Application of discrete event systems for modeling and controlling robotic agents. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2557–2562, 1994.

[9] V. Kumar and K. J. Waldron. Force distribution in walking vehicles on uneven terrain. *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, 112(1):90–99, 1990.

[10] S. J. Lederman and R. L. Klatzky. Hand movements: A window into haptic object recognition. *Cognitive Psychology*, 19:342–368, 1987.

[11] Y. Nakamura, K. Nagai, and T. Yoshikawa. Dynamics and stability in coordination of multiple robotic mechanisms. *International Journal of Robotics Research*, 8(2):44–61, 1989.

[12] P.J. Ramadge and W.M. Wonham. Supervisory control of a class of discrete event processes. *SIAM Journal of Control and Optimization*, 25(1):206–230, 1987.

[13] P.J. Ramadge and W.M. Wonham. The control of discrete-event systems. *Proceedings of the IEEE*, 77(1):81–98, 1989.

[14] K. Rudie and W.M. Wonham. Think globally, act locally: Decentralized supervisory control. *IEEE Transactions on Automatic Control*, 37(11):1692–1708, 1992.

[15] T.M. Sobh and R. Bajcsy. Autonomous observation under uncertainty. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1792–1798, 1992.

[16] J. S. Son, A. Monteverde, and R. D. Howe. A tactile sensor for localizing transient events in manipulation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1994.

[17] S. A. Stansfield. Haptic perception with an articulated, sensate robot hand. Technical report, Sandia National Laboratories, March 1990.

[18] Y. Zheng and J. Y. S. Luh. Optimal load distribution for two industrial robots handling a single object. In *Proceedings of the 1988 International Conference on Robotics and Automation*, pages 344–349, Philadelphia, Pennsylvania, April 1988.

Assignment Program Model for Control of Infinite State Systems

Ratnesh Kumar
Dept. of Elec. Eng.
University of Kentucky
Lexington, KY 40506-0046
kumar@engr.uky.edu

Vijay K. Garg
Dept. of Elec. & Comp. Eng.
University of Texas at Austin
Austin, TX 78712-1084
vijay@pine.ece.utexas.edu

Invited Paper

Abstract

In this paper we study supervisory control of discrete event systems using state variables for representation and specification. The motivation is two fold: firstly, a state variable *representation* allows a concise characterization of systems with *infinitely many states*, and secondly, state variable *specification* allows characterization of qualitative properties of general *nondeterministic* systems. An *assignment program* consisting of state variables and a finite set of conditional assignment statements is used for representing a discrete event system, and a set of forbidden states is used for representing a control specification. The control synthesis problem is undecidable in the general setting. However, in the special case when a single uncontrollable event is present, we show that the problem of computing a maximally permissive supervisor reduces to that of solving an arithmetic equation. Also, in another special case when the system can be represented as a *vector addition system* or equivalently, a Petri net, and the set of forbidden states is a *right-closed* set we provide techniques for maximally permissive supervision. This generalizes the existing results on controlled vector addition systems.

1 Introduction

There has been little work on synthesis of supervisors for general discrete event systems which have infinitely many states and are possibly nondeterministic. In this paper we present a state variable approach to supervisory control of such systems.

One of the advantages of this approach is that a state variable representation allows a concise description of an infinite state system, possibly nondeterministic. We use an *assignment program* consisting of a finite set of state variables for describing the states of the system and a finite set of conditional assignment statements for representing its state transitions. Each assignment statement has a condition or a *guard* associated with it. When the condition is *true*, the assignment statement can *execute* and assign certain new values to the state variables. Thus the system evolves as the assignment statements execute and the value of the state variables change. Such a modeling formalism is general enough to represent any computable discrete event system, i.e., it is Turing machine equivalent [4]. Similar formalism has been used in computer science for modeling and verification [1, 2].

A second advantage of the state variable approach is that a state variable specification can be used to characterize qualitative properties of systems that may be nondeterministic. Most work on supervisory control of discrete event systems use a formal language of event traces to specify desirable system behavior. Such language based specifications can adequately capture the properties of deterministic discrete event systems where the next state is uniquely determined by the current state and the event that occurs in that state. However, when the system is nondeterministic, event trace based specification is not adequate and more general ways of specifying its properties are needed. For example in a nondeterministic system, execution of a certain event trace may result in a set of states some of which are desirable and others undesirable. Then one cannot specify whether such a event trace is desirable or undesirable. Such ambiguity is avoided if state traces rather than event traces are used for specifying system behaviors, since a state trace corresponds to a unique execution sequence.

Temporal logical [9] over state variables can be used to represent desirable state traces by specifying the "safety" and "progress" constraints. Safety refers to properties which assert that certain conditions must *never* occur, whereas progress refers to properties which assert that certain conditions must occur *eventually*. Thus safety imposes constraints on finite length traces of the system, whereas progress constrains the infinite length traces. In this paper, we only consider constraints on finite length traces of the system, i.e., only safety constraints are considered. Specifically, a safety constraint is specified in terms of a set of forbidden states, i.e., only those state traces are desirable that never visit a forbidden state. This is also referred as the state avoidance problem.

In this paper we use *predicates* over state variables to represent a forbidden state specification and use *predicate transformers* [2] to compute supervisors for state avoidance. Predicate transformers have earlier been used in [10, 6, 7] for supervisory control of discrete event plants. However, algorithmic computation of maximally permissive supervisors has not been addressed in those references.

We show that the supervisory synthesis problem for state avoidance is in general undecidable. However, in the special case when there is only one uncontrollable event in the system, or when it is possible to combine all assignment statements corresponding to the uncontrollable events into a single one, then the problem of computing a maximally permissive supervisor reduces to that of solving an arithmetic equation.

Another interesting special case arises when (i) the system is a Petri net, in which case the conditional assignment statements take a simple "linear displacement" form, and (ii) the set of forbidden states is a right-closed set, i.e., if a certain Petri net marking is forbidden, then any other marking *covering* it is also forbidden. We show that the supervisory synthesis problem for state avoidance is *decidable* for such systems and provide an algorithmic technique to compute a maximally permissive supervisor. This generalizes the existing results on controlled Petri nets where supervisory synthesis for state avoidance of right-closed markings (typically specified as the simultaneous marking of certain combination of places) has been solved for only a very restricted class of Petri nets—a subclass of marked graphs [3]. It also generalizes the existing results on controlled vector discrete event systems, which are precisely the controlled Petri nets, where the problem has been solved under the restriction of "loop-freedom" and the set of forbidden states specified as a "linear-inequality predicate" [8], which is also a right-closed set.

2 Notation and Preliminaries

A discrete event system, denoted G, is a 4-tuple $G := (X, \Sigma, \rightsquigarrow, X_0)$, where X denotes the state set, Σ is the finite event set, $\rightsquigarrow \subseteq X \times \Sigma \times X$ is the set of state transitions, and $X_0 \subseteq X$ is the set of initial states. We use state variables to represent the states and a finite set of conditional assignment statements to represent the state transitions. The desired behavior is specified as a predicate over the state variables representing a set of forbidden states.

The notation v is used to denote the vector of state variables of G. If G is n-dimensional, then $v = [v_1, \ldots, v_i, \ldots, v_n]$, where v_i is the ith state variable. The state space X of G equals the cartesian product of domains of all state variables, i.e., $X := \prod_{i=1}^{n} \mathcal{D}(v_i)$, where $\mathcal{D}(v_i)$ is the domain of v_i. By definition, $\mathcal{D}(v_i)$ is a countable set and can be identified with the set of natural numbers $\mathcal{N}$.

We use *predicates* for describing various subsets of the state space. Let $\mathcal{P}(v)$ denote the collection of predicates defined using the state variable vector v, i.e., if $P(v) \in \mathcal{P}(v)$, then it is a boolean valued map $P(v) : X \rightarrow \{0, 1\}$. Consider for example a two dimensional state space $X = \mathcal{Z}^2$. Then the predicate $[v_1 \geq v_2]$ refers to all the states in which the value of variable v_1 is at least as large as the value of variable v_2.

The symbols *true* and *false* are used for denoting predicates that hold on all and none of the states respectively, i.e.,

$$\forall x \in X : true(x) = 1; \quad false(x) = 0.$$

With every predicate $P(v) \in \mathcal{P}(v)$, we associate a set $X_P \subseteq X$ on which $P(v)$ takes the value one. We say that the predicate $P(v)$ holds on $\hat{X} \subseteq X$ if $\hat{X} \subseteq X_P$. Thus the collection of predicates $\mathcal{P}(v)$ has a one-to-one correspondence with the power set 2^X, and the names predicates and subsets can be used interchangeably.

Given $P(v) \in \mathcal{P}(v)$, its *negation* is denoted by $\neg P(v)$. Given an indexing set Λ such that $P_\lambda(v) \in \mathcal{P}(v)$ for each $\lambda \in \Lambda$, the *conjunction* and *disjunction* over Λ is denoted by $\bigwedge_{\lambda \in \Lambda} P_\lambda(v)$ and $\bigvee_{\lambda \in \Lambda} P_\lambda(v)$ respectively. The quadruple $(\mathcal{P}(v), \neg, \wedge, \vee)$ forms a boolean algebra that is isomorphic to the algebra of the subsets of X with the operations of complementation, intersection, and union. For predicates $P(v)$ and $Q(v)$, we say that $P(v)$ is *stronger* than $Q(v)$ (equivalently, $Q(v)$ is *weaker* than $P(v)$), denoted $P(v) \preceq Q(v)$, if $P(v) \wedge Q(v) = P(v)$, or equivalently, $P(v) \vee Q(v) = Q(v)$. It is easily verified that $(\mathcal{P}(v), \preceq)$ is a *complete Boolean lattice* [5].

State transitions map a state to another state. Such mappings are extended from states to set of states or predicates in a natural way, and are known as *predicate transformers*. We use $\mathcal{F}$ to denote the collection of all predicate transformers, i.e., if $f \in \mathcal{F}$, then $f : \mathcal{P}(v) \to \mathcal{P}(v)$.

The *negation* $\neg f$ for some $f \in \mathcal{F}$ is defined by taking the negation of the image predicate. The *conjunction* and *disjunction* of an arbitrary set of predicate transformers is defined in an analogous way and is obtained by taking the conjunction and disjunction respectively over the set of image predicates. The *conjunctive closure* of f, denoted f_*, and *disjunctive closure* of f denoted f^* are defined to be $\bigwedge_{i \geq 0} f^i$ and $\bigvee_{i \geq 0} f^i$ respectively, where f^0 is the identity predicate transformer and $f^{i+1} := f(f^i)$.

Given $f : X \to X$, the *substitution predicate transformer* "$v := f(v)$" maps a predicate $P(v)$ to $P(f(v))$. Consider for example the f given by

$$v_1, v_2 := v_1 + v_2, v_1 - v_2.$$

Then the corresponding substitution predicate transformer maps the predicate $[v_1 < v_2]$ to $[v_1 + v_2 < v_1 - v_2] = [v_2 < 0]$.

3 Assignment Program Model

Here we describe the assignment program model for G. The state variables of G is defined using a declaration of the type:

declare
v_1 : integer; $v_2 : \{0, \ldots, N\}$;

Here the system has two state variables v_1 and v_2 with $\mathcal{D}(v_1) = \mathcal{Z}$ and $\mathcal{D}(v_2) = \{0, \ldots, N\}$. The declare section also lists the event labels of the system.

The initial state set of G is specified as an *initial predicate*, denoted $I(v)$, using a declaration statement of the type:

initially
$I(v)$,

where $I(v)$ is such that $X_I = X_0$, the set of initial states of G.

The state transitions "$\rightsquigarrow$" of G is specified using a finite set of conditional assignment statements of the form:

assign
$\sigma : C_\sigma(v) \to v := f_\sigma(v)$

where $\sigma \in \Sigma$ is an event, $C_\sigma(v)$ is a predicate, called the *guard*, and $f_\sigma : X \to X$ is a map defined on the state space. If no guard is present, then *true* is treated as the guard. A conditional assignment statement of the above type is *enabled* if the condition $C_\sigma(v)$ holds. An enabled assignment statement may *execute*. Upon execution, new values are assigned to the state variables according to the map f_σ and a state transition on the event σ occurs. Consider for example an assignment statement:

$$\sigma : [v_1 < v_2] \to v_1, v_2 := v_2, v_1.$$

It is enabled when value of v_1 is smaller than the value of v_2, and upon execution swaps the values of the two variables.

The following example illustrates the representation of a discrete event system as an assignment program.

Example 1 Consider a *Readers_Writers* system which has a distributed database access to which is sought by infinitely many readers and writers. The variables *nr* and *nw* denote the number of readers and writers respectively, and the symbols *st_rd*, *st_wt*, *end_rd*, *end_wt* denote the events start-read, start-write, end-read, end-write respectively.

```
plant Readers_Writers
declare
$nr, nw$ : integer
$st\_rd, st\_wt, end\_rd, end\_wt$ : event
initially
$nr, nw = 0, 0$
assign
  $st\_rd$ :                              $nr := nr + 1$
  $st\_wt$ :                              $wr := wr + 1$
  $end\_rd$ :   $[nr - 1 \geq 0]$   $\to$   $nr := nr - 1$
  $end\_wt$ :   $[nw - 1 \geq 0]$   $\to$   $nw := nw - 1$
end Readers_Writers
```

The substitution predicate transformer can be used to define the *forward one-step reachable*, *fr*, and *backward one-step reachable*, *br*, predicate transformers for G. Intuitively, *fr* determines the "postcondition" after the occurrence of a state transition for a given starting condition, whereas *br* determines the "precondition" prior to the occurrence of a state transition for a given resulting condition. This is illustrated in Figure 1.

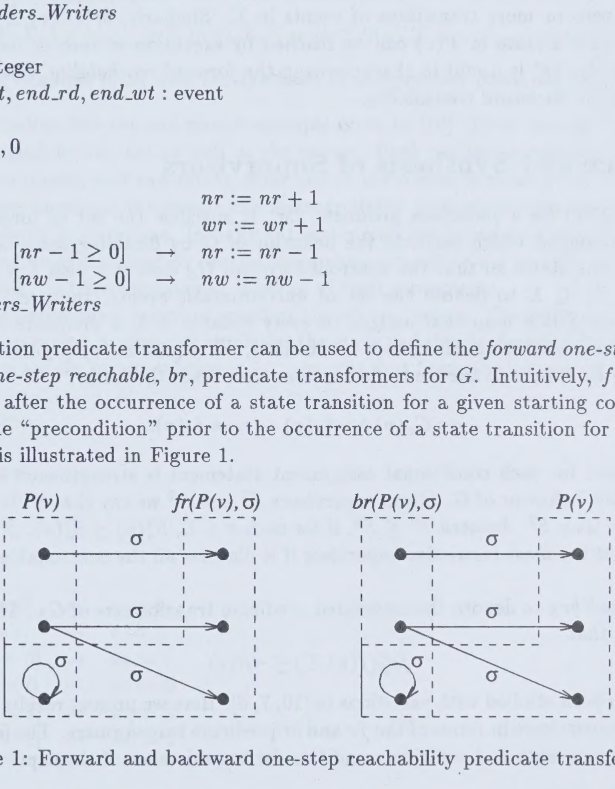


Figure 1: Forward and backward one-step reachability predicate transformers

For the assignment statement $\sigma : C_\sigma(v) \to v := f_\sigma(v)$ and a condition $P(v)$, these are formally defined as follows:

$$fr(P(v), \sigma) := C_\sigma(f_\sigma^{-1}(v)) \wedge P(f_\sigma^{-1}(v)); \quad br(P(v), \sigma) := C_\sigma(v) \wedge P(f_\sigma(v)).$$

Stated another way, if prior to the execution of the assignment statement, the system state x is one where $C_\sigma(v) \wedge P(v)$ holds, then the resulting state must have the value $f_\sigma(x)$, i.e., $C_\sigma(f_\sigma^{-1}(v)) \wedge$

$P(f_\sigma^{-1}(v))$ holds at the resulting state, which is the postcondition. Similarly, if after the execution of the assignment statement, the system state x is one where $P(v)$ holds, then $C_\sigma(v)$ must hold at the starting state x' and $f_\sigma(x') = x$, i.e, $C_\sigma(v) \wedge P(f_\sigma(v))$ holds at the starting state, which is the precondition. Note that the computation of br is easier as compared to that of fr, since its computation does not require the computation of f^{-1}, which is often impossible to compute.

The following example illustrates fr and br computations.

Example 2 Suppose the state space is $\mathcal{Z}^2$, and there is a single assignment statement, $\sigma : v_1, v_2 := v_1 - v_2, v_1 + v_2$. Let $P(v) = [v_1 \leq v_2]$. Then

$$fr(P(v), \sigma) = [f_\sigma^{-1}(v_1) \leq f_\sigma^{-1}(v_2)] = [\frac{v_1 + v_2}{2} \leq \frac{v_2 - v_1}{2}] = [v_1 \leq 0].$$

Similarly,

$$br(P(v), \sigma) = [v_1 - v_2 \leq v_1 + v_2] = [v_2 \geq 0].$$

For $\hat{\Sigma} \subseteq \Sigma$, we define $fr(P(v), \hat{\Sigma}) := \bigvee_{\sigma \in \hat{\Sigma}} fr(P(v), \sigma)$, and $br(P(v), \hat{\Sigma}) := \bigvee_{\sigma \in \hat{\Sigma}} br(P(v), \sigma)$. Finally, note that $fr^*(P(v), \hat{\Sigma})$ denotes the set of states which are reachable from a state in $P(v)$ by execution of zero or more transitions of events in $\hat{\Sigma}$. Similarly, $br^*(P(v), \hat{\Sigma})$ denotes the set of states from where a state in $P(v)$ can be reached by execution of zero or more transitions of events in $\hat{\Sigma}$. Clearly, fr^* is useful in characterizing the *forward reachability*, whereas br^* is useful in characterizing the *backward reachability*.

4 Existence and Synthesis of Supervisors

Let $B(v) \in \mathcal{P}(v)$ be a *forbidden predicate*, i.e., it specifies the set of forbidden states. A supervisor S is designed which restricts the behavior of G by disabling some of its controllable events at its various states so that the controlled system G_S does not visit any of the forbidden states. We use $\Sigma_u \subseteq \Sigma$ to denote the set of *uncontrollable* events; the events in $\Sigma - \Sigma_u$ are *controllable*. Then S is a map that assigns to every event $\sigma \in \Sigma$ a predicate $S_\sigma(v)$ where σ is disabled by S. Since uncontrollable cannot be disabled, we require $S_\sigma(v) = false$ for each $\sigma \in \Sigma_u$. Each assignment statement of the controlled system then takes the form:

$$\sigma : C_\sigma(v) \wedge \neg S_\sigma(v) \rightarrow v := f_\sigma(v).$$

Since the guard for each conditional assignment statement is strengthened in G_S, it is clear that S restricts the behavior of G. Given supervisors S^1 and S^2 we say that S^1 is *more permissive* or *less restrictive* than S^2, denoted $S^1 \leq S^2$, if for each $\sigma \in \Sigma$, $S^1_\sigma(v) \preceq S^2_\sigma(v)$. S is said to be the least permissive or the most restrictive supervisor if it disables all the controllable events in all the states.

We use fr_S and br_S to denote the associated predicate transformers of G_S. The control task is to design S such that

$$fr^*_S(I(v), \Sigma) \preceq \neg B(v).$$

This problem has been studied with variations in [10, 7, 6]. Here we present results for the existence and synthesis of supervisors in terms of the fr and br predicate transformers. The following theorem provides a useful necessary and sufficient condition for the existence of the supervisor:

Theorem 1 Given a plant G and a forbidden predicate $B(v)$, there exists S such that $fr^*_S(I(v), \Sigma) \preceq \neg B(v)$ if and only if $I(v) \preceq \neg br^*(B(v), \Sigma_u)$.

Let $\mathcal{S}$ be the family of supervisors such that for each $S \in \mathcal{S}, fr^*_S(I(v), \Sigma) \preceq \neg B(v)$. Then it is easy to show that $\mathcal{S}$ contains a maximally permissive supervisor $S^\dagger$ whenever it is nonempty. The following theorem gives an explicit characterization of $S^\dagger$.

Theorem 2 Given a plant G and a forbidden predicate $B(v)$, if $\mathcal{S} \neq \emptyset$, then a maximally permissive supervisor is given by: $S_\sigma^\uparrow(v) = br(br^*(B(v), \Sigma_u), \sigma)$ for any $\sigma \in (\Sigma - \Sigma_u)$.

The test for the existence of a supervisor as well as the computation of a maximally permissive supervisor (when one exists) require the computation of $br^*(B(v), \Sigma_u)$. By definition

$$br^*(B(v), \Sigma_u) = \bigvee_{n \geq 0} br^n(B(v), \Sigma_u).$$

Also, for the assignment statements of the type $\sigma : C_\sigma(v) \to v := f_\sigma(v)$, we have:

$$br(B(v), \Sigma_u) = \bigvee_{\sigma \in \Sigma_u} [C_\sigma(v) \wedge B(f_\sigma(v))].$$

So the following recursion can be used to compute $br^*(B(v), \Sigma_u)$:

- $br^0(B(v), \Sigma_u) = B(v)$
- $br^{n+1}(B(v), \Sigma_u) = \bigvee_{\sigma \in \Sigma_u} [C_\sigma(v) \wedge br^n(B(f_\sigma(v)), \Sigma_u)]$,
- $\exists m : [br^m(B(v), \Sigma_u) \preceq \bigvee_{n<m} br^n(B(v), \Sigma_u)] \Rightarrow [br^*(B(v), \Sigma_u) = \bigvee_{n<m} br^n(B(v), \Sigma_u)]$.

The following example illustrates the synthesis of a supervisor using our technique.

Example 3 We adopt the cat and mouse example given in [10]. There are infinitely many rooms that can be occupied by the cat as well as the mouse. Both cat and mouse can move from room zero to any of the rooms, and can return from any of the rooms to room zero. These motions are controllable. From room one the mouse can uncontrollably go to room two, and from there it can uncontrollably return to room one. Initially cat and mouse are in rooms zero and one respectively. A supervisor must be synthesized to control the movements of the cat and the mouse so that they never share a room, and have maximum freedom of their movement.

The following assignment program specifies the above system. The state variable c is used to denote the room occupied by the cat; similarly, the state variable m denotes the room occupied by the mouse. There are six events labeled $\sigma_1, \ldots, \sigma_6$, the first four of which are controllable.

```
plant maze
declare
  c, m : integers
  σ1, ..., σ6 : events
initially
  c, m = 0, 1
assign
  σ1 :  [c = 0]  →  c := i
  σ2 :  [c = i]  →  c := 0
  σ3 :  [m = 0]  →  m := i
  σ4 :  [m = i]  →  m := 0
  σ5 :  [m = 1]  →  m := 2
  σ6 :  [m = 2]  →  m := 1
end maze
```

In this example $X = \mathcal{N}^2$ and the forbidden state specification is given by $B(c,m) = [c = m]$. Note that there are infinitely many states and also that the system is nondeterministic since can and mouse can nondeterministically move to any of the rooms from room zero. In order to verify the existence of a supervisor we check whether $I(c,m) \preceq \neg br^*(B(c,m), \Sigma_u)$. We use the recursive computation given above to compute $br^*(B(c,m), \Sigma_u)$.

1. $br^0(B(c,m),\Sigma_u)$
   $= [c = m]$.

2. $br^1(B(c,m),\Sigma_u)$
   $= [br((c = m), \sigma_5)] \vee [br((c = m), \sigma_6)]$
   $= [(m = 1) \wedge (c = 2)] \vee [(m = 2) \wedge (c = 1)]$.

3. $br^2(B(c,m),\Sigma_u)$
   $= [br([(m = 1) \wedge (c = 2)] \vee [(m = 2) \wedge (c = 1)], \sigma_5)]$
   $\vee [br([(m = 1) \wedge (c = 2)] \vee [(m = 2) \wedge (c = 1)], \sigma_6)]$
   $= [(m = 1) \wedge \{[(2 = 1) \wedge (c = 2)] \vee [(2 = 2) \wedge (c = 1)]\}]$
   $\vee [(m = 2) \wedge \{[(1 = 1) \wedge (c = 2)] \vee [(1 = 2) \wedge (c = 1)]\}]$
   $= [(m = 1) \wedge \{[(false) \wedge (c = 2)] \vee [(true) \wedge (c = 1)]\}]$
   $\vee [(m = 2) \wedge \{[(true) \wedge (c = 2)] \vee [(false) \wedge (c = 1)]\}]$
   $= [(m = 1) \wedge (c = 1)] \vee [(m = 2) \wedge (c = 2)]$
   $\preceq [c = m]$
   $= br^0(B(c,m),\Sigma_u)$.

Hence

$$br^*(B(c,m),\Sigma_u)$$
$$= br^0(B(c,m),\Sigma_u) \vee br^1(B(c,m),\Sigma_u)$$
$$= [c = m] \vee [(c = 1) \wedge (m = 2)] \vee [(c = 2) \wedge (m = 1)].$$

Since $I(c,m) = [(c = 0) \wedge (m = 1)]$, it follows that

$$I(c,m) \wedge br^*(B(c,m),\Sigma_u)$$
$$= [0 = 1] \vee [(0 = 1) \wedge (1 = 2)] \vee [(0 = 2) \wedge (1 = 1)]$$
$$= [false] \vee [(false) \wedge (false)] \vee [(false) \wedge (true)]$$
$$= false.$$

This implies $I(c,m) \preceq \neg br^*(B(c,m),\Sigma_u)$. So it follows from Theorem 1 that a supervisor exists for this case.

We can use the result of Theorem 2 to compute the maximally permissive supervisor:

1. $S_{\sigma_1}(c,m)$
   $= br(br^*(B(c,m),\Sigma_u),\sigma_1)$
   $= [c = 0] \wedge [(i = m) \vee [(i = 1) \wedge (m = 2)] \vee [(i = 2) \wedge (m = 1)]]$
   $= [c = 0] \wedge [m = i]$

2. $S_{\sigma_2}(c,m)$
   $= br(br^*(B(c,m),\Sigma_u),\sigma_2)$
   $= [c = i] \wedge [(0 = m) \vee [(0 = 1) \wedge (m = 2)] \vee [(0 = 2) \wedge (m = 1)]]$
   $= [c = i] \wedge [(m = 0) \vee [(false) \wedge (m = 2)] \vee [(false) \wedge (m = 1)]]$
   $= [c = i] \wedge [m = 0]$

3. $S_{\sigma_3}(c,m)$
   $= br(br^*(B(c,m),\Sigma_u),\sigma_3)$
   $= [m = 0] \wedge [(c = i) \vee [(c = 1) \wedge (i = 2)] \vee [(c = 2) \wedge (i = 1)]]$
   $= [m = 0] \wedge [c = i]$

4. $S_{\sigma_4}(c,m)$
   $= br(br^*(B(c,m),\Sigma_u),\sigma_4)$
   $= [m = i] \wedge [(c = 0) \vee [(c = 1) \wedge (0 = 2)] \vee [(c = 2) \wedge (0 = 1)]]$
   $= [m = i] \wedge [(c = 0) \vee [(c = 1) \wedge (false)] \vee [(c = 2) \wedge (false)]]$
   $= [m = i] \wedge [c = 0]$

Thus when one of them is in a room other than zero, and the other is in room zero, then the later is prevent from leaving room zero (control policy for events σ_1 and σ_3), whereas the former is prevented from moving into room zero (control policy for σ_2 and σ_4).

5 Computation of br^*: General Case

In the previous example the recursive computation of $br^*(B(v), \Sigma_u)$ terminated in a finite number of steps. However, this may not always be the case, and we need systematic methods to decide the termination of the recursive computation. In the present section we study this for a general assignment program, whereas the next section studies the special case of Petri nets. We first show that in general an algorithmic computation of $br^*(B(v), \Sigma_u)$ is not possible by showing that the problem of determining membership in $br^*(B(v), \Sigma_u)$ is in general undecidable.

Theorem 3 For a plant G, and a forbidden predicate $B(v)$, it is in general undecidable to determine whether a given state is in $br^*(B(v), \Sigma_u)$.

5.1 Single Uncontrollable Event

Although the algorithmic computation of $br^*(B(v), \Sigma_u)$ is in general infeasible, its computation can be reduced to solving an arithmetic equation when only a single uncontrollable event is present, or if it is possible to combine multiple conditional assignment statements corresponding to the various uncontrollable events into a single one. This can aide in the computation of $br^*(B(v), \Sigma_u)$ providing a "semi-algorithmic" technique for its computation. We give several examples in this section to illustrate such a "semi-algorithmic" technique.

First note that when there is a single uncontrollable event σ, then

$$br^*(B(v), \Sigma_u) = br^*(B(v), \sigma) = \exists n : n \in \mathcal{N} : br^n(B(v), \sigma),$$

where an existential quantification based notation has been used [2]. In the following theorem we give a formula for $br^n(B(v), \sigma)$ which is used to compute $br^*(B(v), \sigma)$ using the existential quantification notation.

Theorem 4 For all $n \geq 1$, $br^n(B(v), \sigma) = B(f_\sigma^n(v)) \bigwedge_{i \leq n-1} C_\sigma(f_\sigma^i(v))$.

The formula in Theorem 4 requires multiple conjunctions. However, as shown in the following corollary the formula is much simplified when f_σ satisfies certain conditions with respect to $C_\sigma(v)$.

Definition 1 A function $f : X \to X$ is said to be *increasing* with respect to a predicate $P(v) \in \mathcal{P}(v)$ if $P(v) \preceq P(f(v))$; it is said to be *decreasing* with respect to $P(v)$ if $P(f(v)) \preceq P(v)$.

Corollary 1 If f_σ is increasing with respect to $C_\sigma(v)$, then for $n \geq 1$,

$$br^n(B(v), \sigma) = B(f_\sigma^n(v)) \wedge C_\sigma(v).$$

If f_σ is decreasing with respect to $C_\sigma(v)$, then for $n \geq 1$,

$$br^n(B(v), \sigma) = B(f_\sigma^n(v)) \wedge C_\sigma(f_\sigma^{n-1}(v)).$$

As an application of results of this subsection we revisit the cat and mouse problem.

Example 4 In the cat and mouse problem, it is possible to combine the two uncontrollable events into a single uncontrollable event σ:

$$\sigma : [(m = 1) \vee (m = 2)] \to (m := 3 - m).$$

So the technique of this section can be applied for the computation of $br^*(B(c, m), \Sigma_u)$.

6 Computation of br^*: Petri net Case

When G is a Petri net its state space is $X = \mathcal{N}^n$, i.e., it is the set of n-dimensional positive integer valued vectors. We show that in the case of Petri nets the recursive computation of $br^*(B(v), \Sigma_u)$ terminates whenever the set of forbidden states is right-closed.

Definition 2 Given a set of positive integer valued vectors $\hat{X} \subseteq \mathcal{N}^n$, it is said to be *right-closed* if $x \in \hat{X}$ and $x' \geq x$ implies $x' \in \hat{X}$. A predicate $P(v) \in \mathcal{P}(v)$ is said to be right-closed if X_P is a right-closed set.

In the following lemma we show that for a Petri net plant $br^*(B(v), \Sigma_u)$ is right-closed whenever $B(v)$ is right-closed.

Lemma 1 If the plant G has a Petri net model, and the forbidden predicate $B(v)$ is right-closed, then $br^*(B(v), \Sigma_u)$ is also a right-closed predicate.

The above lemma can be used to obtain the following main result of this section. It uses the fact that a right-closed set of positive integer valued vectors possesses a *finite* set of minimal elements.

Theorem 5 For a Petri net plant G and a right-closed forbidden predicate $B(v)$, the recursive computation of $br^*(B(v), \Sigma_u)$ terminates in a finite number of iterations.

Remark 1 Since the computation of $br^*(B(v), \Sigma_u)$ does not depend on the conditional assignment statements corresponding to the controllable events, the result of Theorem 5 is applicable in a weaker setting when the *uncontrollable sub-system*, i.e., the system described by the conditional assignment statements corresponding to the uncontrollable events, can be represented as a Petri net.

In order to illustrate the result of Theorem 5 we revisit the *Readers_Writers* example.

Example 5 Consider the *Readers_Writers* assignment program of Example 1. Clearly it represents a Petri net. Note that the *st_rd* and *st_wt* events are always enabled, i.e., they are the so called *source* transitions of the net. Suppose it is required that at most one writer is allowed access to the database at any time, and when a writer is accessing the database no reader should be given access to the database. In other words, the forbidden predicate is give by:

$$B(nr, nw) = [nw \geq 2] \vee [(nw = 1) \wedge (nr \neq 0)].$$

It is easy to see that $B(nr, nw)$ is a right-closed predicate with two minimal states: $nr, nw = 0, 2$ and $nr, nw = 1, 1$. So it follows from Theorem 5 that the recursive computation of $br^*(B(nr, nw), \Sigma_u)$ will terminate in a finite number of iterations. Refer to [6, Example 4.6] where the computation of the maximally permissive supervisor for this example has been discussed in detail.

7 Conclusion

We have examined the suitability of using assignment programs to model discrete event systems, and shown that this model is general enough to represent any computable discrete event system. We have shown that automatic supervisory synthesis for state avoidance is in general infeasible. We believe that techniques need to be developed so that the synthesis can be done with some human assistance. To this end, we have shown that in case there is a single uncontrollable event, or if it is possible to express multiple uncontrollable events as a single conditional assignment statement, then the supervisory synthesis problem for state avoidance can be reduced to the problem of solving arithmetic equations. We have obtained conditions under which a closed form formula can be obtained. Also, we show that when the system can be represented as a Petri net and the forbidden state set is right-closed, such as when it is specified as simultaneous marking of certain combination of places, then automatic supervisory synthesis is possible. The problem of determining an upper bound on the number of iterations needed to compute the supervisor is a challenging problem for future investigation.

We have investigated the supervisory control of infinite state systems for state avoidance. Such desired specifications are "static" in the sense that they can be enforced by *static* supervisors. However, when the desired specification is specified as more general constraints on the state traces, such as the cat should never visit a room more than twice, then more general *dynamic* supervisors will be needed. Automatic design of such supervisors for infinite state systems is an interesting problem for further investigation. This will require developing a suitable temporal logic for specifying useful constraints on finite length state traces, and techniques for computing maximally permissive supervisors to enforce such constraints.

Acknowledgement

This research was supported in part by the Center for Robotics and Manufacturing, University of Kentucky, in part by the National Science Foundation under Grants NSF-ECS-9409712, NSF-ECS-9414780, and NSF-CCR-9110605, and in part by a TRW faculty assistantship award.

References

[1] K. M. Chandy and J. Misra. *Parallel Program Design: A Foundation.* Addison-Wesley, Reading, MA, 1988.

[2] E. W. Dijkstra and C. S. Scholten. *Predicate Calculus and Program Semantics.* Springer-Verlag, New York, 1990.

[3] L. E. Holloway and B. H. Krogh. Synthesis of feedback control logic for a class of controlled Petri nets. *IEEE Transactions on Automatic Control*, 35(5):514–523, May 1990.

[4] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley, Reading, MA, 1979.

[5] R. Kumar and V. K. Garg. *Modeling and Control of Logical Discrete Event Systems.* Kluwer Academic Publishers, Boston, MA, 1995.

[6] R. Kumar, V. K. Garg, and S. I. Marcus. Predicates and predicate transformers for supervisory control of discrete event systems. *IEEE Transactions on Automatic Control*, 38(2):232–247, February 1993.

[7] Y. Li and W. M. Wonham. Control of vector discrete event systems I - the base model. *IEEE Transactions on Automatic Control*, 38(8):1214–1227, August 1993.

[8] Y. Li and W. M. Wonham. Control of vector discrete event systems II - controller synthesis. *IEEE Transactions on Automatic Control*, 39(3):512–531, 1994.

[9] A. Pnueli. The temporal logic of programs. In *Proceedings of 18th Annual Symposium on Foundations of Computer Science*, pages 46–57, Providence, RI, November 1977.

[10] P. J. Ramadge and W. M. Wonham. Modular feedback logic for discrete event systems. *SIAM Journal of Control and Optimization*, 25(5):1202–1218, 1987.

Petri Net Supervisors for DES in the Presence of Uncontrollable and Unobservable Transitions[1]

John O. Moody and Panos J. Antsaklis
Department of Electrical Engineering
University of Notre Dame, Notre Dame, IN 46556

Abstract

This paper describes a computationally efficient method for synthesizing feedback controllers for plants modeled by Petri nets which may contain uncontrollable transitions. The controller, a Petri net itself, computed using the concept of Petri net place invariants, enforces a set of linear constraints on the plant. It is shown how the original set of plant behavioral constraints can be transformed into a new set that will yield a controller which enforces the original constraints without directly influencing any of the uncontrollable transitions. Previous researchers have shown that this constraint transformation can be performed by solving a linear integer programming problem; the method presented in this paper is simpler computationally, but, depending on the structure of the uncontrollable part of the plant, maximal permissiveness may not be guaranteed.

1 Introduction

A methodology to automatically derive feedback controllers described by a Petri net was presented in [4]. It is assumed that the DES plant is described by an ordinary Petri net. The method is based on the idea that specifications representing desired plant behaviors can be enforced by making them invariants of the controlled net. The resulting controllers are identical to the monitors [2] of Giua et al. and consist only of places and arcs. The controller's size is proportional to the number of constraints.

This approach has been recently extended to apply to Petri nets which contain uncontrollable and unobservable transitions, the firing of which cannot be controlled or cannot be observed respectively. This extension is also based on place invariants and it is again easy and transparent to implement with excellent numerical properties. These results on Petri nets with uncontrollable transitions complement and are compared with results on uncontrollable vector DES recently reported in the literature [3].

This invariants based approach to design of Petri net feedback controllers has a number of advantages: (a) the design method is transparent as it is based on the concept of place invariants, (b) the resulting controller and consequently the overall controlled system are described by ordinary Petri nets where a variety of tools for analysis both graphical and algebraic are available. Verification of the design is therefore rather straightforward. (c) The design method has excellent numerical properties. Although it is based on the concept of invariants, it is not actually necessary to calculate any of these invariants. The controller design involves only a multiplication of the incidence matrix by a vector representing the constraints to be imposed on the system. This make this control design approach particularly appealing in control reconfiguration applications where because of a failure the controller must be redesigned on line.

The paper is structured as follows. The controller synthesis method [2, 4] for plants with controllable transitions is reviewed in section 2. The new approach for transforming constraints in the face of uncontrollable transitions and generating the corresponding controller is presented in section 3. Concluding remarks are given in section 4.

---

[1] The research for this paper has been supported in part by the Center for Applied Mathematics at the University of Notre Dame.

2 Automatic Controller Synthesis from Plant Constraints

The system to be controlled is modeled by a Petri net with n places and m transitions and is known as the plant or *process net*. The incidence matrix of the process net is D_p. It is assumed that all the enabled transitions can fire. It is possible that the process net will violate certain constraints placed on its behavior, thus the need for control. The *controller net* is a Petri net with incidence matrix D_c made up of the process net's transitions and a separate set of places. The *controlled system* or *controlled net* is the Petri net with incidence matrix D made up of both the original process net and the added controller. The control goal is to force the process to obey constraints of the following form

$$L\mu_p \leq b \tag{1}$$

where μ_p is the marking vector of the Petri net modeling the process, L is an $n_c \times n$ integer matrix, b is an n_c dimensional integer vector and n_c is the number of constraints. Note that the inequality is with respect to the individual elements of the two vectors $L\mu_p$ and b and can be thought of as the logical conjunction of the individual "less than or equal to" constraints. This definition will be used throughout this paper whenever vectors appear on either side of an inequality sign.

Suppose we wish to enforce the single constraint $\mu_1 + \mu_2 \leq 1$ which means that at most one of the two places p_1 and p_2 can be marked, or, in other words, both places cannot be marked at the same time. This inequality constraint can be transformed into an equality by introducing an external Petri net controller which contains a place which represents a nonnegative "slack variable" μ_c. The constraint then becomes $\mu_1 + \mu_2 + \mu_c = 1$ or, in general

$$L\mu_p + \mu_c = b \tag{2}$$

where μ_c is an n_c dimensional integer vector which represents the marking of the controller places. Note that $\mu_c \geq 0$ because the number of tokens in a place can not become negative; thus equation (2) implies inequality (1). The controller places insure that the weighted sums of tokens in the process net's places are always less than or equal to the elements of b. The places which maintain the inequality constraints are part of a separate net called the controller net. The structure of the controller net will be computed by observing that the introduction of the slack variables forces a set of place invariants on the overall controlled system defined by equation (2).

A place invariant is defined as every integer vector x which satisfies

$$x^T \mu = x^T \mu_0 \text{ (a constant)} \tag{3}$$

where μ_0 is the net's initial marking, and μ represents any subsequent marking. Equation (3) means that the weighted sum of the tokens in the places of the invariant remains constant at all markings and this sum is determined by the initial marking of the Petri net. The place invariants of a net are elements of the kernel of the net's incidence matrix, i.e., they can be computed by finding integer solutions to

$$x^T D = 0 \tag{4}$$

where D is an $n \times m$ incidence matrix with n being the number of places and m the number of transitions.

The matrix D_c contains the arcs that connect the controller places to the transitions of the process net. Let $\mathbb{Z}$ be the set of integers. The incidence matrix $D \in \mathbb{Z}^{(n+n_c)\times m}$ of the closed loop system is given by

$$D = \begin{bmatrix} D_p \\ D_c \end{bmatrix} \tag{5}$$

and the marking vector $\mu \in \mathbb{Z}^{n+n_c}$ and initial marking μ_0 are given by

$$\mu = \begin{bmatrix} \mu_p \\ \mu_c \end{bmatrix} \qquad \mu_0 = \begin{bmatrix} \mu_{p_0} \\ \mu_{c_0} \end{bmatrix} \tag{6}$$

Note that equation (2) is in the form of (3), thus the invariants defined by equation (2) on the system (5), (6) must satisfy equation (4).

$$\begin{aligned} X^T D = [L \ \ I] \begin{bmatrix} D_p \\ D_c \end{bmatrix} &= 0 \\ LD_p + D_c &= 0 \end{aligned} \tag{7}$$

where I is an $n_c \times n_c$ identity matrix since the coefficients of the slack variables in equation (2) are all equal to 1.

Proposition 1. The Petri net controller, $D_c \in \mathbb{Z}^{n_c \times m}$ with initial marking μ_{c_0}, which enforces constraints (1) when included in the closed loop system (5) with marking (6) is defined by

$$D_c = -LD_p \tag{8}$$

with initial marking

$$\mu_{c_0} = b - L\mu_{p_0} \tag{9}$$

assuming that the transitions with input arcs from D_c are controllable.

Proof. Equation (8) is simply the solution of equation (7) which forces $\begin{bmatrix} L & I \end{bmatrix}$ to be invariants of the closed loop system. Since $\begin{bmatrix} L & I \end{bmatrix}$ are invariants we known that $L\mu_p + \mu_c = L\mu_{p_0} + \mu_{c_0}$ and from equation (9), $L\mu_{p_0} + \mu_{c_0} = b$. The marking of a place is always greater than or equal to zero, therefore

$$\begin{aligned} L\mu_p + \mu_c &= b \\ L\mu_p &\le b \end{aligned}$$

□

Remark. Proposition 1 creates a controller which will enable and inhibit various transitions in the plant. If any of these transitions are uncontrollable then the controller defined by this method is invalid. The next section shows how a transformation of the constraints can be performed so that the uncontrollable transitions in the net receive no input arcs from the controller places.

Remark. Since our method admits the structure of the process net as well as a set of specifications, it can control transitions that participate in self-loops in the process net. This is because the constraints on these transitions are part of the specifications. Note that when an element of D_c is zero, there are no arcs at all connecting the given place and transition, i.e., there are no canceling self-loops in the controller structure.

Remark. The controller defined by proposition 1 is maximally permissive, assuming controllable transitions, in that it will never disable a transition that would directly violate the constraints if fired. The proof of this result is given in [4].

3 Handling Uncontrollable Transitions

Consider the situation where the controller is not allowed to influence certain transitions in the plant Petri net. These transitions are called uncontrollable. It is illegal for the Petri net controller to include an arc from one of the controller places to any of these uncontrollable plant transitions.

Let D_u be the incidence matrix of the uncontrollable portion of the process net. D_u is composed of the columns of D_p which correspond to the uncontrollable transitions. Recall that, assuming no self loops, positive elements in an incidence matrix refer to arcs from transitions to places, and negative elements refer to arcs from places to transitions. $D_u \in \mathbb{Z}^{m \times n_u}$ where n_u is the number of uncontrollable transitions. Given a set of constraints, $L\mu \leq b$, the Petri net controller given by $D_c = -LD_p$ violates the uncontrollability constraint if LD_u contains any elements greater than zero. The uncontrollability constraint dictates that we can not draw any arcs from the controller places to the transitions, and the portion of the controller corresponding to the uncontrollable transitions is given by $-LD_u$, thus the elements of LD_u must all be less than or equal to zero.

3.1 Controller Computation

Suppose we are unable to meet the uncontrollability constraint and have positive values in the matrix LD_u. It is necessary to transform the constraint vector L such that the original constraint of $L\mu_p \leq b$ is still maintained, while obeying the uncontrollability constraint. A proposition is given later in this section which shows how to construct a controller which meets both of these conditions. The proposition is supported by the lemma below.

Lemma 2. If

$$L'\mu_p \leq b' \tag{10}$$

where

$$\begin{aligned} L' &= R_1 + R_2 L && (11)\\ b' &= R_2(b + \mathbf{1}) - \mathbf{1} && (12) \end{aligned}$$

$R_1 \in \mathbb{Z}^{n_c \times m}$, $R_2 \in \mathbb{Z}^{n_c \times n_c}$, $\mathbf{1}$ is an n_c dimensional vector of 1's and

$$R_1\mu_p \geq 0 \qquad \text{for all possible } \mu_p \tag{13}$$

$$R_2 \text{ is a positive definite diagonal matrix} \tag{14}$$

then $L\mu_p \leq b$.

Proof. The transformed constraint is

$$\begin{aligned} L'\mu_p &\leq b' \\ (R_1 + R_2 L)\mu_p &\leq R_2(b + \mathbf{1}) - \mathbf{1} \end{aligned}$$

because all of the elements are integers, the inequality can be transformed into a strict inequality:

$$(R_1 + R_2 L)\mu_p < R_2(b + \mathbf{1})$$

and because R_2 is diagonal and positive definite,

$$R_2^{-1} R_1 \mu_p + L\mu_p < b + \mathbf{1}$$

Assumptions (13) and (14) imply that all elements of the vector $R_2^{-1} R_1 \mu_p$ must be greater than or equal to zero, therefore

$$L\mu_p < b + 1$$

and, once again because the inequality deals strictly with integers, we have

$$L\mu_p \leq b$$

□

According to proposition 1, the incidence matrix of the controller which will enforce (10) is given by $D_c = -L'D_p$. In order to meet the uncontrollability constraint we need $L'D_u \leq 0$ which will insure that the controller contains no arcs leading to the uncontrollable transitions in the plant. Then

$$R_1 D_u + R_2 L D_u \leq 0 \quad \text{or}$$

$$\begin{bmatrix} R_1 & R_2 \end{bmatrix} \begin{bmatrix} D_u \\ LD_u \end{bmatrix} \leq 0 \tag{15}$$

Note that if $L'D_u = 0$ then the rows of the matrix $\begin{bmatrix} R_1 & R_2 \end{bmatrix}$ lie within the kernel of $\begin{bmatrix} D_u \\ LD_u \end{bmatrix}$.

A trivial solution to inequality (15) is given by $\begin{bmatrix} R_1 & R_2 \end{bmatrix} = \begin{bmatrix} -L & I \end{bmatrix}$. However this choice for R_1 will either violate assumption (13) or dictate an impossible control task. If $L\mu$ has both positive and negative elements then we can not say that $R_1\mu = -L\mu$ is always greater than zero and assumption (13) is violated. If $L\mu$ is always positive or zero then $R_1\mu = -L\mu$ violates assumption (13) for arbitrary μ. If $L\mu$ is always negative or zero and $b \geq 0$ then the constraints will always be met by the plant without control, and if $L\mu$ is always negative or zero and $b \leq 0$ then a choice of $R_1\mu = -L\mu$ will create a transformed constraint of $0 \leq b$ which is meaningless since b is a constant vector.

Lemma 2 may now be combined with proposition 1 to synthesize a Petri net controller for a plant with uncontrollable transitions.

Proposition 3. Given a plant Petri net with incidence matrix D_p, a set of uncontrollable transitions, a set of linear constraints $L\mu \leq b$ on the net marking, and R_1 and R_2 which meet assumptions (13) and (14) and satisfy inequality (15), then there exists a controller with incidence matrix

$$D_c = -(R_1 + R_2 L) D_p \tag{16}$$

such that if

$$\mu_{c_0} = R_2(b + 1) - 1 - (R_1 + R_2 L)\mu_{p_0} \tag{17}$$

is a valid initial marking, i.e. $\mu_{c_0} \geq 0$, then the constraint $L\mu_p \leq b$ will be satisfied by the closed loop system. This controller contains no input arcs to any of the uncontrollable transitions.

Proof. Note that, according to proposition 1, equations (16) and (17) define a controller that enforces the constraint $L'\mu \leq b'$ where $L' = R_1 + R_2 L$ and $b' = R_2(b+1) - 1$. Lemma 2 shows that if assumptions (13) and (14) are met then a controller which enforces a particular constraint $L'\mu \leq b'$ will also enforce the constraint $L\mu \leq b$. Because R_1 and R_2 satisfy inequality (15), no controller arcs are drawn to the uncontrollable transitions.
□

Remark. If any element of μ_{c_0} calculated according to equation (17) is negative then the corresponding constraint is impossible to enforce. This is the situation where only trivial solutions to the constraint transformation exist because the uncontrollable portion of the plant is incompatible with a particular control goal.

The usefulness of proposition 3 lies in whether or not it is possible to find R_1 and R_2 which meet the necessary assumptions. To meet assumption (13), it is sufficient to assume that all of the elements of R_1 are nonnegative, since the elements of μ_p are nonnegative by definition. In general, for unbounded μ_p, it is necessary that all of the elements of R_1 be nonnegative, however if bounds on μ_p are known, then it is possible to generate valid R_1 vectors which contain some negative elements. If R_1 and R_2 which satisfy (13) and (14) do exist, then they can be found by performing row operations on $\begin{bmatrix} D_u \\ LD_u \end{bmatrix}$. Row operations act as premultiplications of a matrix, just as $\begin{bmatrix} R_1 & R_2 \end{bmatrix}$ premultiplies $\begin{bmatrix} D_u \\ LD_u \end{bmatrix}$ in inequality (15). R_1 and R_2 can be found by finding a set of row operations which do not involve premultiplication of any row by a negative number and which force the LD_u portion of the matrix to contain all zero or negative elements. This operation is relatively easy to perform. Note that assumption (14), which requires R_2 to be a positive definite matrix, is not restrictive. This matrix simply represents the premultiplication coefficients of the rows of the LD_u portion of the matrix undergoing row operations. We can assume this matrix is diagonal because LD_u is linearly dependent with D_u, i.e., we will never need to take linear combinations of the rows in LD_u. We can also assume that the diagonal elements are positive since, if negative numbers are required, they can be accounted for by R_1, which still needs to meet assumption (13).

It is possible that uncontrollable transitions in a plant might make a particular constraint impossible to realize. In this case it may still be possible to find R_1 and R_2 such that they meet the assumptions, however the transformed constraint $L'\mu_p \leq b'$ will be trivial. For example, consider the Petri net in figure 1. Suppose that we wish to limit the number of tokens that enter p_2, i.e., the untransformed constraint is $\mu_2 \leq b$. If the single transition is uncontrollable, then we will obtain a transformed constraint of $\mu_1 + \mu_2 \leq b$, which is already the case if there are b or fewer tokens in the net and is impossible if the net starts with more than b tokens.

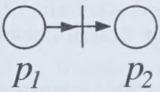


Figure 1: A Petri net that yields a trivial constraint transformation.

3.2 Example

We now provide a simple example in order to illustrate the concepts that have been covered above. The example plant is partially based on the model of an "unreliable machine" from [1]. The machine is used to process parts from an input queue, completed parts are moved to an output queue. The machine is considered unreliable because it is possible that it may break down and damage a part during operation. This behavior is captured in the plant model. Damaged parts are moved to a separate queue from the queue for successfully completed parts. The Petri net model of the plant is shown in figure 2, and a description of the various places and transitions is given in table 1.

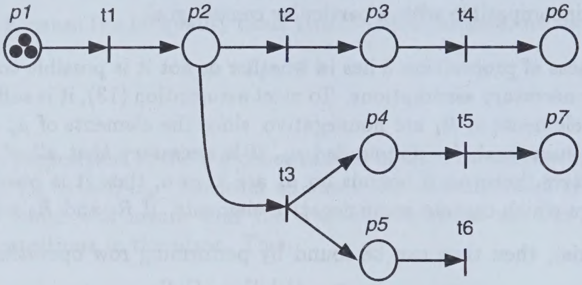


Figure 2: Petri net model of an uncontrolled unreliable machine.

Places

p_1	Input queue - Number of tokens = parts remaining
p_2	Machine is "up and busy," part is being processed
p_3	Part is waiting for transfer to completed parts queue
p_4	Part is waiting for transfer to damaged parts queue
p_5	Machine is waiting to be repaired
p_6	Completed parts queue
p_7	Damaged parts queue

Transitions

t_1	Part moves from input queue to machine
t_2	*Uncontrollable:* Part processing is complete
t_3	*Uncontrollable:* Machine fails, part is damaged
t_4	Part moves to completed parts queue
t_5	Part moves to damaged parts queue
t_6	Machine is repaired

Table 1: Place and transition descriptions for the Petri net of figure 2.

The plant model has two uncontrollable transitions, t_2 and t_3. Transition t_3 represents machine break down and so obviously can not be controlled. Transition t_2 is considered uncontrollable because the controller can not force the machine to instantly finish a part that is not yet completed, nor does it direct the machine to stop working on an unfinished part. The transition is labeled uncontrollable in order to prevent a control design from attempting either of these two actions.

The Petri net model of the plant has the following incidence matrix and marking vector.

$$D_p = \left[\begin{matrix} \underbrace{\begin{matrix} -1 & 0 & 0 \\ 1 & -1 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix}}_{D_u} & \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{matrix} \end{matrix}\right] \qquad \mu_p = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \\ \mu_7 \end{bmatrix} \tag{18}$$

The initial conditions are $\mu_{p_0} = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$.

If the machine is broken, we do not want to load a new part until repairs have been completed. This means that places p_2 and p_5 should contain at most one token:

$$\mu_2 + \mu_5 \leq 1 \tag{19}$$

Parts waiting to be transferred to a storage queue, whether completed or damaged, wait in the same position on the machine. The Petri net model uses two places, p_3 and p_4, to represent waiting parts, because there are two different destinations. In order to prevent conflict, the second constraint is

$$\mu_3 + \mu_4 \leq 1 \tag{20}$$

Using the matrix form of constraint (1) we have

$$\underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}}_{L} \mu_p \leq \underbrace{\begin{bmatrix} 1 \\ 1 \end{bmatrix}}_{b} \tag{21}$$

First we must check the uncontrollability condition.

$$LD_u = \begin{bmatrix} -1 & 0 \\ 1 & 1 \end{bmatrix}$$

We need all of the elements of LD_u to be less than or equal to zero if we are to avoid using uncontrollable transitions. There is no problem with the first row, but a transformation will have to be found to eliminate the 1's in the second row. This can be done by applying row operations from the matrix D_u to eliminate the positive elements in the second row of LD_u.

Because constraint (19) required no transformation, the first row of R_1 will be all zeros. A row operation involving the addition of the second row of the D_u matrix was required to transform constraint (20), thus the second row of R_1 will be all zeros with a one in the second column. It was not necessary to premultiply either constraint, thus R_2 will be an identity matrix.

$$R_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad R_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

We now apply equations (11) and (12) to find the transformed constraints represented by L' and b'.

$$\underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}}_{L'} \mu_p \leq \underbrace{\begin{bmatrix} 1 \\ 1 \end{bmatrix}}_{b'}$$

The controller is the calculated using equations (16) and (17).

$$D_c = -L'D_p = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

$$\mu_{c_0} = b' - L'\mu_{p_0} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

The controlled net is shown in figure 3. The constraint logic is enforced and no input arcs are drawn to the uncontrollable transitions.

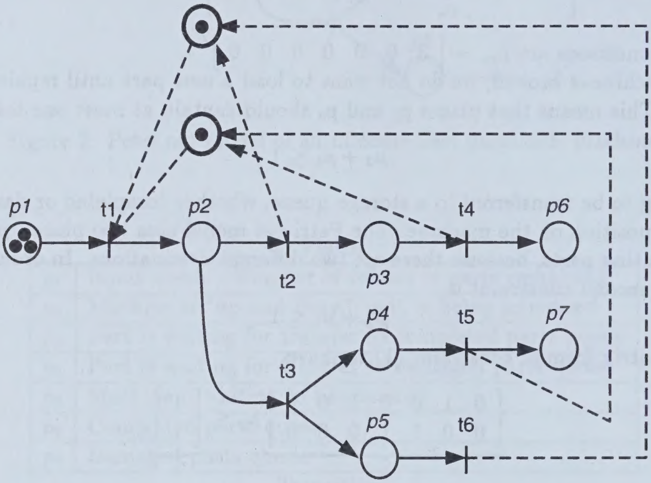


Figure 3: The controlled unreliable machine.

3.3 Discussion

An extensive look at many of the issues central to this research can be found in the work of Li and Wonham [3]. These authors show that optimal, or maximally permissive, control actions which account for uncontrollable transitions can be found by repeated applications of a linear integer programming problem (LIP), assuming that valid control actions actually exist and that the uncontrollable portion of the net contains no loops. They also give sufficient conditions under which the solution to the LIP has a closed form expression. These conditions place a certain tree structure on the uncontrollable portion of the net. When this tree structure is further limited, Li and Wonham are able to prove that the optimal control law which insures $L\mu_p \leq b$ can be written $C\mu_p \leq d$. This is the case where it is possible to represent the action of the optimal control law with ordinary Petri nets. In this situation, it is possible to find R_1 and R_2 by performing row operations on $\begin{bmatrix} D_u \\ LD_u \end{bmatrix}$ which is much more desirable, computationally, than analytically solving an LIP. However the tree structure assumed by Li and Wonham is only sufficient, not necessary, for example, the structure of the uncontrollable part of the plant in section 3.2 does not conform to Li and Wonham's "type 2 tree structure," however an optimal solution was found and implemented using an ordinary Petri net controller. There are also cases where, following the procedures presented above, suboptimal Petri net controllers may be derived. These suboptimal controllers may be sufficient for many tasks, depending on the application. Further exploration and codification of these issues is one of the main goals of this ongoing research.

4 Conclusions

This paper has presented a particularly simple method for constructing feedback controllers for untimed Petri nets, even in the face of uncontrollable plant transitions. The method is based on the idea that specifications representing desired plant behaviors can be enforced by making them invariants of the controlled net, and that simple row operations on a matrix containing the uncontrollable columns of the plant incidence matrix can be used to eliminate controller use of illegal transitions.

The significance of this particular approach to Petri net controller design is that the control net can be computed very efficiently, thus the method shows promise for controlling large, complex systems, or for recomputing the control law online due to some plant failure.

Besides the problem of uncontrollable plant transitions, there are also situations where it is not possible to observe certain transitions. For example, it may be too difficult or costly to place a sensor in a certain part of a plant, or a sensor may fail which might require the fast online computation of a new control law. Because of a duality in the mathematics between uncontrollable and unobservable transitions, this research extends itself naturally to the problem of unobservable transitions.

References

[1] A. A. Desrochers and R. Y. Al-Jaar, *Applications of Petri Nets in Manufacturing Systems*, IEEE Press, Piscataway, NJ, 1995.

[2] A. Giua, F. DiCesare, and M. Silva, “Generalized Mutual Exclusion Constraints on Nets with Uncontrollable Transitions”, *Proceedings of the 1992 IEEE International Conference on Systems, Man, and Cybernetics*, Chicago, IL, pp. 974–979, October 1992.

[3] Y. Li and W. M. Wonham, “Control of Vector Discrete-Event Systems II – Controller Synthesis,” *IEEE Transactions on Automatic Control,* 39(3), March 1994.

[4] J. O. Moody, K. Yamalidou, M. Lemmon, and P. J. Antsaklis, “Feedback Control of Petri Nets Based on Place Invariants”, *Proceedings of the 33rd Conference on Decision and Control*, Vol. 3, pp 3104–3109, Lake Buena Vista, FL, December 1994.

[5] P. J. G. Ramadge and W. M. Wonham, “The Control of Discrete Event Systems,” *Proceedings of the IEEE*, vol 77, No. 1, pp 81–97, January 1989.

[6] W. Reisig, *Petri Nets*, Springer-Verlag, 1985.

[7] W. M. Wonham and P. J. G. Ramadge, “On the Supremal Controllable Sublanguage of a Given Language”, *SIAM J. Control Optim.*, vol. 25, no. 3 pp 637-659, 1987.

Multiobjective Control via Infinite-Dimensional LMI Optimization.

Nicola Elia* Peter M. Young† Munther A. Dahleh‡

Abstract

Many practical performance specifications are equivalent to linear or convex constraints on the closed loop map. In these cases, the design of closed loop systems that achieve the desired level of performance becomes a convex optimization problem that can be cast in terms of Linear Matrix Inequalities. However, these problems are often infinite dimensional. The key to tackling such problems with finite computation tools is to have a primal-dual formulation of the problem without duality gap. In this paper, we present a survey of results on duality theory for linearly constrained infinite dimensional optimization problems and their extension to LMI optimization problems. These results can be applied to several multiobjective control problems, and in particular to ℓ_1 optimal control problems subject to a variety of LMI constraints.

1 Introduction

Over the last decade, optimal control theory has seen advances from the classical $\mathcal{H}_2$ theory (see [8]) with the solution of norm minimization problems such as $\mathcal{H}_\infty$ [6] and ℓ_1 [7]. Although these different norms allow the designer to emphasize different criteria in the design, and hence exploit different tradeoffs in the problem, it is not always possible to capture all the desired performance specifications in a single norm constraint. This has motivated the research for mixed norm minimization problems, like $\mathcal{H}_2/\mathcal{H}_\infty$ [9, 10], $\ell_1/\mathcal{H}_\infty$ [11], $\ell_1/\mathcal{H}_2$ [12], and $\mathcal{H}_2/\ell_1$ [13].

The use of Linear Matrix Inequality (LMI) constraints has received much attention in the literature in recent years, mainly because this general class of constraints can capture many interesting control specifications (see [14] and the references therein), and because Finite LMI problems are readily computable.

However, the LMI problems encountered in optimal and robust control theory are often infinite dimensional (for instance when the primal variables represent the impulse response of a discrete time system). In that case, the key to developing a tractable solution is to have a primal-dual formulation of the problem without duality gap. From such a formulation it is possible to uncover important structures in the problem that may allow one to solve it exactly. However, we must point out that, in the cases where a finite dimensional solution to the problem cannot be found, it is important that both primal and dual problems can be approximated arbitrarily well by finite dimensional optimizations.

In this paper we present a survey of duality results for infinite dimensional LMI optimization problems. In particular, in section 3, we present some sufficient conditions for the lack of a duality gap for infinite dimensional LMI's, and give an exact characterization of the primal and

---

*Lab. for Info. and Decision Systems, M.I.T., 35-417, Cambridge, MA 02139, U.S.A., Tel: 617-253 2346, e-mail: nicola@lids.mit.edu

†Department of Electrical Engineering, Colorado State University, Fort Collins, CO 80523, U.S.A., Tel: 970-491 5406, e-mail: pmy@lance.colostate.edu

‡Lab. for Info. and Decision Systems, M.I.T., 35-402, Cambridge, MA 2139, U.S.A., Tel: 617-253 3892, e-mail: dahleh@lids.mit.edu

dual problems. To illustrate the utility of these tools, in section 4 we consider three problems from optimal control theory . Firstly we address the problem of minimizing the ℓ_1 norm (of a closed loop system) subject to bounds on the frequency response magnitude at a finite number of points. Secondly, we consider the mixed $\ell_1/\mathcal{H}_2$ problem, i.e., minimizing the ℓ_1 norm subject to a bound on the $\mathcal{H}_2$ norm. This problem was considered in [12], where the authors developed a solution only to the SISO (or square MIMO) case. Finally, we consider the robust $\mathcal{H}_2$, or mixed $\mathcal{H}_2/\ell_1$, For all the problems we present the exact primal-dual formulation with no duality gap for the general (nonsquare) MIMO case. A simple numerical example is computed in detail for each problem.

Although these are interesting results, the real power of the tools presented here lies in the generality of the LMI description. Thus, similar results can be derived for multiobjective problems with many combinations of linear, norm and LMI constraints.

2 Notation and Preliminaries

Note that a real valued sequence $\{x_n\}_{n=0}^{\infty}$ can always be associated with an infinite vector x where x_n is the corresponding element of x. Thus we make no distinction between the two and write $x = \{x_n\}_{n=0}^{\infty}$. For such a vector x we define the following norms:

$$\|x\|_p \doteq \sqrt[p]{\sum_{n=0}^{\infty} |x_n|^p} \qquad \text{for } 1 \le p < \infty, \qquad \|x\|_\infty \doteq \sup_{n \ge 0} |x_n|$$

We define the corresponding Banach spaces $\ell_p \doteq \{x : \|x\|_p < \infty\}$ We denote $\mathbb{R}_S^{m \times m}$ as the space of all (square) symmetric matrices in $\mathbb{R}^{m \times m}$, and for $S \in \mathbb{R}_S^{m \times m}$ the notation $S \ge 0$ means that S is positive semidefinite. We refer to this as a matrix inequality, and we take $S \ge Q$ to mean $(S - Q) \ge 0$.

Recall that an LMI constraint is one of the form $\qquad \mathcal{S}(x) \ge S^C$ where $\mathcal{S}$ depends linearly on x, and the above notation means that $\mathcal{S}(x) - S^C$ is positive semidefinite (we assume throughout that $\mathcal{S}, S^C$ are symmetric, without loss of generality). A special class of LMI is the class of finite size LMI i.e., $\mathcal{S}(x), S^C \in \mathbb{R}_S^{m \times m}$. Linear vector constraints, (LP constraints), of the form $Ax \ge b$ with $A : X \to \mathbb{R}^N$, fit into this category.

Note that many LMI constraints can be recast as a single LMI constraint, i.e.,

$$\begin{array}{l} \mathcal{S}_1(x) \ge S_1^C \\ \mathcal{S}_2(x) \ge S_2^C \end{array} \longleftrightarrow \text{block diag}\,(\mathcal{S}_1(x), \mathcal{S}_2(x)) \ge \text{block diag}\,(S_1^C, S_2^C)$$

Thus, we may effectively incorporate any mix of LP and LMI constraints into a single LMI constraint.

We now introduce some basic definitions for infinite size LMI's. We consider two dimensional sequences $\{S_{ij}\}_{i,j=1}^{\infty}$, which we associate with the square matrix S, whose $(i,j)^{\text{th}}$ element is S_{ij}. This is written as $S = [S_{ij}]$. For such a matrix we denote the induced $\ell_2 - \ell_2$ norm by $\|S\|_{2i}$ and we define

$$\ell_{2i} = \{S : \|S\|_{2i} < \infty\} \tag{1}$$

For a finite matrix we have $\|S\|_{2i} = \overline{\sigma}(S)$, where $\overline{\sigma}$ denotes maximum singular value. We denote $\text{Tr}(S)$ as $\text{Tr}(S) \doteq \sum_{i=1}^{\infty} S_{ii} \doteq \lim_{n \to \infty} \left(\sum_{i=1}^{n} S_{ii} \right)$ and it is well defined whenever the above limit exists. The transpose is defined as usual as $[S_{ij}]^T \doteq [S_{ji}]$

In the case that S is a finite symmetric matrix, then the definition of an LMI constraint of the form $S \ge 0$ is equivalent to: $\quad v^T S v = \sum_{i,j=1}^{m} S_{ij} v_i v_j \ge 0 \quad \forall \quad v \in \mathbb{R}^m$

In the case that S is a symmetric and infinite size matrix then a natural extension of this

definition is that $S \geq 0 \longleftrightarrow v^T S v \geq 0 \;\forall\; v \in \ell_2$
where the notation $v^T S v$ is shorthand for $v^T S v \doteq \sum_{i,j=1}^{\infty} S_{ij} v_i v_j \doteq \lim_{m\to\infty} \left(\sum_{i,j=1}^{m} S_{ij} v_i v_j\right)$. In order for this definition to make sense we require that the above limit exists. To this end we make the following definition.

Definition 2.1 *An LMI space is a linear vector space whose elements, S, are square symmetric matrices. For infinite size matrices we require that $v^T S v$ converges and $v^T S v < \infty$ for all $v \in \ell_2$.*

We will refer to elements of an LMI space as LMI matrices. For these matrices the LMI constraint $S \geq 0$ is well defined. Note that any square symmetric matrix $S \in \ell_{2i}$ is an LMI matrix ([5]). Moreover it is shown in [5] that the set of square symmetric matrices $S \in \ell_{2i}$, equipped with the norm $\|S\| \doteq \|S\|_{2i}$ is a LMI Banach space, i.e., a complete LMI space.

3 Duality Theory Results for LMI Problems

We will be concerned with constrained convex optimization problems. In particular, we will consider problems of the following form:

$$\begin{aligned} &\inf_{x \in X} f(x) \\ &\mathcal{S}(x) \geq Q \end{aligned} \tag{2}$$

for some Banach space X, where $\mathcal{S}(x), Q$ are symmetric, and we require $f(x)$ and $\mathcal{S}(x)$ to be linear. We refer to these problems as Linear Matrix Inequality problems (LMI). Much recent research has been aimed at the efficient solution of finite dimensional LMI's, with the result that powerful new tools are emerging (see [14] and the references therein).

3.1 Dual Spaces and Duality Theory

In the following we assume that X, Y are Banach spaces. The notation $\mathcal{S} \in \mathcal{B}(X, Y)$ means that $\mathcal{S}$ is a bounded linear operator from X to Y, i.e., $\mathcal{S} : X \to Y$ is a linear operator, and the induced operator norm $\|\mathcal{S}\| \doteq \sup_{\|x\|_X \leq 1} \|\mathcal{S}(x)\|_Y$ is finite.
The dual space X^* is defined as the space of all bounded linear functionals on X, i.e., $X^* = \mathcal{B}(X, \mathbb{R})$. The dual norm is taken to be the induced operator norm, and with this norm X^* is a Banach space. For an element $x^* \in X^*$, we denote its operation on an element $x \in X$ by $\langle x, x^* \rangle$. Note that, once we have a definition of a nonnegative cone in X, i.e., the set $P = \{x \in X : x \geq 0\}$ is defined, then this defines a nonnegative cone in the dual space via:

$$P^{\oplus} \doteq x^* \geq 0 \quad \longleftrightarrow \quad P^{\oplus} = \{\langle x, x^* \rangle \geq 0 \;\forall\; x \in P\} \tag{3}$$

The Lagrangian duality theory establishes a duality correspondence between the primal problem in the space X and a dual problem in the dual of the constraint space. The constraint space is the space where the image of the constraint operator lies. The optimal dual cost is always smaller or equal to the optimal primal cost. In some cases one can guarantee that in fact this lower bound is exact, i.e., the primal and dual problems share the same objective value. In this case we say that there is no duality gap.

In general, if one allows infinite dimensional LP's or LMI's, then a duality gap can exist. Unfortunately, many interesting robust control analysis and synthesis problems lead to infinite dimensional optimization problems (see the next section). For this reason, it is important to characterize in general those infinite dimensional problems for which there is no duality gap. Next, we present some results that guarantee the lack of duality gaps for different convex optimization problems.

Next theorem is the most general duality result presented in the paper. Theorems for LMI problems can be derived form it by an opportune choice of positive cones.

Theorem 3.1 *[16] Let X, Z be Banach spaces and P be a cone in X given by the intersection of a finite number of cones P_i in X, $P = \cap_{i=1}^{n} P_i$. Assume that P contains an interior point and that there exists an $x_1 \in X$ such that $\mathcal{A}x_1 = b$ and $-x_1$ is an interior point of P. Assume that $\mathcal{A} \in \mathcal{B}(X, Z)$ has closed range and that the optimal value μ^o of the optimization is finite. Then,*

$$\mu^o = \begin{array}{ll} \inf\limits_{x \in X} & \langle x, c \rangle \\ \mathcal{A}x = b & \\ -x \in P & \end{array} = \begin{array}{ll} \max\limits_{z^* \in Z^*, \lambda \in X^*} & \langle b, z^* \rangle \\ -\mathcal{A}^* z_1^* + c + \lambda = 0 & \\ \lambda \in \sum_{i=1}^{n} P_i^{\oplus} & \end{array}$$

and the maximum is achieved for some $z_o^* \in Z^*$ and $\lambda \in \sum_{i=1}^{n} P_i^{\oplus} \subset X^*$. If the infimum on the left is achieved by some $x_o \in X$, then, $\langle x_o, c \rangle = \langle x_o, \mathcal{A}^* z_o^* \rangle$

In many cases we are interested in constrained norm minimization problems. The following duality result specializes Theorem 3.1, for minimum norm problem with LMI constraints.

Theorem 3.2 *[5] Let X, Z be Banach spaces, Y an LMI Banach space, $b \in Z, S^C \in Y$, and $\mathcal{A} \in \mathcal{B}(X, Z), \mathcal{S} \in \mathcal{B}(X, Y)$. Suppose that $\mathcal{A}$ has closed range in Z, the nonnegative cone in Y contains an interior point, and there exists $x \in X$ such that $\mathcal{A}x = b$ and $\mathcal{S}(x) > S^C$. Then it follows that*

$$\begin{array}{l} \inf\limits_{x \in X} \|x\| \\ \mathcal{A}x = b \\ \mathcal{S}(x) \geq S^C \end{array} = \begin{array}{l} \max\limits_{z^* \in Z^*, y^* \in Y^*} \langle b, z^* \rangle + \langle S^C, y^* \rangle \\ \|\mathcal{A}^* z^* + \mathcal{S}^* y^*\| \leq 1 \\ y^* \geq 0 \end{array} \tag{4}$$

4 Applications

We now present an application of these new tools to three controller synthesis problems. The first two problems we study are: ℓ_1 minimization with $\mathcal{H}_\infty$ constraints and ℓ_1 minimization with $\mathcal{H}_2$ constraints. The last problem we study is a $\mathcal{H}_2$ minimization with ℓ_1 constraints. Although this problem is not naturally posed as an LMI problem, it can be shown that it is equivalent to an appropriate LMI problem.

4.1 Mixed ℓ_1/LMI Performance Problem

The ℓ_1 norm is of particular interest since it can be rewritten as a linear objective (via the introduction of slack variables). Note that, provided we do not have interpolation zeros on the unit circle, the primal problems are readily put in the form of theorem 3.2 where the operator $\mathcal{A} : \ell_1^{n_z \times n_w} \to \ell_1$ has closed range and n_z and n_w are the number of outputs and inputs of the closed loop system. See [15] for details.

In order to get a concrete representation of the dual formulation we need to assume more about the Banach spaces Y, Z, namely we need bases for Y, Z so that we can use these to induce bases for the dual spaces Y^*, Z^*. This is particularly simple when the constraints are of finite size (which is often what we are interested in). Let the operators $\mathcal{A}$ and $\mathcal{S}$ be represented as:

$$\mathcal{A}\phi = \sum_{n=0}^{\infty} \phi(n) a(n), \qquad \mathcal{S}(\phi) = \sum_{n=0}^{\infty} \phi(n) S(n)$$

Theorem 4.1 *Let $b \in \mathbb{R}^l$, $S^C \in \mathbb{R}_S^{m \times m}$ and $\mathcal{A} \in \mathcal{B}(\ell_1, \mathbb{R}^l), \mathcal{S} \in \mathcal{B}(\ell_1, \mathbb{R}_S^{m \times m})$. Suppose that there exists $\phi \in \ell_1$ such that $\mathcal{A}\phi = b$ and $\mathcal{S}(\phi) > S^C$. Then it follows that*

$$\begin{array}{l} \inf\limits_{\phi \in \ell_1} \|\phi\|_1 \\ \mathcal{A}\phi = b \\ \mathcal{S}(\phi) \geq S^C \end{array} = \begin{array}{l} \max\limits_{\gamma \in \mathbb{R}^l, \Gamma \in \mathbb{R}_S^{m \times m}} \gamma^T b + \mathrm{Tr}(\Gamma S^C) \\ |\gamma^T a(n) + \mathrm{Tr}(\Gamma S(n))| \leq 1 \\ \Gamma \geq 0 \end{array} \tag{5}$$

Note that this result is *not* covered by the well known duality results for finite LMI's (see, e.g., [14]). Although the LMI constraint is of finite size (i.e., $\mathcal{S}(\phi) \in \mathbb{R}_S^{m\times m}$), the LMI is not finite dimensional (i.e., $\mathcal{S}(\phi)$ is a function of $\phi \in \ell_1$ with infinitely many terms).

Note that In this case both the primal and the dual take the form of LMI optimization problems.

4.2 ℓ_1 Minimization with $\mathcal{H}_\infty$ Constraints

We now impose a constraint on the magnitude of the frequency response of the closed loop system at a frequency point ω_0 (or a finite number of such points). This problem was considered in [17], where the authors developed an approximate solution using a LP approach. From the definition of the Fourier transform $\hat{\Phi}(e^{j\omega}) = \sum_{n=0}^{\infty} \Phi(n) e^{j\omega n}$ the evaluation of the frequency response at a point, i.e., $\hat{\Phi}(e^{j\omega_0})$ for fixed ω_0, is a linear functional of Φ. By Schur lemma [5, 14], the $\mathcal{H}_\infty$ constraint $\overline{\sigma}(\hat{\Phi}(e^{j\omega_0})) \leq \beta$ can be recast as an LMI constraint

$$\begin{pmatrix} \beta I & \hat{\Phi}(e^{j\omega_0}) \\ \hat{\Phi}^*(e^{j\omega_0}) & \beta I \end{pmatrix} \geq 0$$

LMI's on complex Hermitian matrices can be rearranged to LMI's on real symmetric matrices ([5]). If Φ has n_z outputs and n_w inputs, then this LMI lies in $\mathbb{R}_S^{2(n_z+n_w)\times 2(n_z+n_w)}$, which is finite dimensional; hence the nonnegative cone has an interior point. Thus, provided we have a strictly feasible point, i.e., there exists a stabilizing LTI controller for which $\overline{\sigma}(\hat{\Phi}(e^{j\omega_0})) < \beta$, we satisfy the requirements of theorem 3.2 and can apply (4) to yield the exact primal-dual formulation of the problem without duality gap. To summarize, we have the following result:

Theorem 4.2 *Consider a general ℓ_1 minimization problem, with no interpolation zeros on the unit circle, subject to a finite number of $\mathcal{H}_\infty$ constraints: $\overline{\sigma}(\hat{\Phi}(e^{j\omega_k})) \leq \beta_k$ for $k = 1, \ldots, K$ as described above. Then, provided there exists a stabilizing LTI controller satisfying $\overline{\sigma}(\hat{\Phi}(e^{j\omega_k})) < \beta_k$ for $k = 1, \ldots, K$ this problem has no duality gap and has the exact primal-dual formulation in (4).*

The following simple numerical example illustrates the approach. Consider the following non-minimum phase unstable plant

$$\hat{P}(\lambda) = \frac{\lambda - \frac{1}{2}}{\lambda + \frac{2}{3}}. \tag{6}$$

We first solve a standard ℓ_1 sensitivity minimization problem: $\nu_1 = \inf_{K-\text{stabilizing}} \|(1 - PK)^{-1}\|_1$ We obtain the the optimal cost is $\nu_1 = \frac{10}{7} = 1.429$ and the optimal solution is: $\hat{\phi}(\lambda) = \frac{4}{7} + \frac{6}{7}\lambda$.
Now consider the magnitude of the frequency response at $\omega_0 = \frac{\pi}{2}$ for this solution $|\hat{\phi}(e^{j\frac{\pi}{2}})| = 1.03$
We can now augment this optimization problem with the frequency response constraint $|\hat{\phi}(e^{j\frac{\pi}{2}})| \leq 0.8$. This constraint can be written as an LMI, as outlined above. Defining

$$S(n) = \begin{bmatrix} 0 & \cos(n\omega_0) & 0 & -\sin(n\omega_0) \\ \cos(n\omega_0) & 0 & \sin(n\omega_0) & 0 \\ 0 & \sin(n\omega_0) & 0 & \cos(n\omega_0) \\ -\sin(n\omega_0) & 0 & \cos(n\omega_0) & 0 \end{bmatrix}$$

and $S^C = -0.8 I_4$, the new primal problem becomes:

$$\nu_2 = \inf_{\phi \in \ell_1} \|\phi\|_1$$

subject to: $\hat{\phi}(-\frac{2}{3}) = 0$, $\hat{\phi}(\frac{1}{2}) = 1$, and $\mathcal{S}(\phi) \geq S^C$

Applying theorem 4.1, we find that the *exact* dual to this problem is given as

$$\nu_2 = \max_{\substack{\gamma_1, \gamma_2 \in \mathbb{R} \\ \Gamma \in \mathbb{R}_S^{4\times 4}}} \gamma_1 - 0.8\mathrm{Tr}(\Gamma$$

subject to: $|\gamma_1(\frac{1}{2})^n + \gamma_2(-\frac{2}{3})^n + \mathrm{Tr}(\Gamma S(n))| \leq 1$ and $\Gamma \geq 0$
Only the constraints for $n = 0, 1, 3$ are active in the dual problem. The optimal cost for this problem is $\nu_2 = 1.587$ and the optimal solution is: $\hat{\phi}(\lambda) = 0.5841 + 0.7748\lambda + 0.2281\lambda^3$.

Although this problem involved an infinite dimensional LMI, by using the duality results we were able to solve it *exactly* with a finite LMI. In fact, for SISO (or square MIMO) problems, the solution will typically be FIR, and hence computable exactly with a finite LMI, although it is possible to construct SISO examples for which this is not true. For such problems however one can still get upper and lower bounds as finite LMI's which converge to the optimal solution.

4.3 The Mixed $\ell_1/\mathcal{H}_2$ Problem

In this section we consider imposing a constraint on the $\mathcal{H}_2$ norm of the closed loop system, namely

$$\|\Phi\|_{\mathcal{H}_2} = \sqrt{\sum_{i=1}^{n_z}\sum_{j=1}^{n_w} \|\Phi_{ij}\|_2^2} \leq \beta \tag{7}$$

The authors considered this problem in [12] and they developed a solution only to the SISO (or square MIMO) case.

Note that it is trivial to arrange the entries of Φ to generate an infinite vector ϕ so that

$$\|\Phi\|_{\mathcal{H}_2} = \|\phi\|_2 \tag{8}$$

Thus we need to impose the constraint $\|\phi\|_2 \leq \beta$. This constraint is equivalent to the infinite size LMI

$$\begin{pmatrix} \beta & \phi^T \\ \phi & \beta I \end{pmatrix} \geq 0 \tag{9}$$

One can now check that this is an infinite dimensional LMI Banach space, and the non-negative cone contains an interior point (choose $\beta \in \mathbb{R}$ and $\phi \in \ell_1$ for which $\|\phi\|_2 < \beta$). Once again the only additional assumption we need in order to apply theorem 3.2 is that the constraint is strictly feasible, i.e., there exists a stabilizing LTI controller for which $\|\Phi\|_{\mathcal{H}_2} < \beta$. To summarize, we have the following result:

Theorem 4.3 *Consider a general ℓ_1 minimization problem, with no interpolation zeros on the unit circle, subject to an $\mathcal{H}_2$ constraint: $\|\Phi\|_{\mathcal{H}_2} \leq \beta$ as described above. Then, provided there exists a stabilizing LTI controller satisfying $\|\Phi\|_{\mathcal{H}_2} < \beta$, this problem has no duality gap and has the exact primal-dual formulation in (4).*

Consider once again the numerical example from section 4.2. Recall that the pure ℓ_1 optimal controller achieved $\|\phi\|_1 = \frac{10}{7} = 1.429$ and one may readily compute that for this solution $\|\phi\|_2 = 1.03$. On the other hand, if one solves for a pure $\mathcal{H}_2$ optimal controller, it is easy to check that the solution yields $\|\phi\|_1 = 1.8413$ and $\|\phi\|_2 = \frac{4\sqrt{3}}{7} = 0.9897$.

Let's now solve for a controller minimizing the ℓ_1 norm of the closed loop system subject to the additional $\mathcal{H}_2$ constraint

$$\|\phi\|_2 \leq 1$$

(which is close to the $\mathcal{H}_2$ optimal). The primal problem becomes:

$$\nu_3 = \inf_{\phi \in \ell_1} \|\phi\|_1$$

subject to $\hat{\phi}(-\frac{2}{3}) = 0$, $\hat{\phi}(\frac{1}{2}) = 1$, and $\begin{pmatrix} 1 & \phi^T \\ \phi & I \end{pmatrix} \geq 0$

Applying theorem 3.2, we find that the *exact* dual to this problem is given as

$$\nu_3 = \max_{\substack{\gamma_1,\gamma_2,\delta \in \mathbb{R} \\ \psi \in \ell_2}} \gamma_1 - \delta$$

subject to $|\gamma_1(\frac{1}{2})^n + \gamma_2(-\frac{2}{3})^n + \psi_n| \leq 1$ and $\begin{pmatrix} \delta & \psi^T \\ \psi & \delta I \end{pmatrix} \geq 0$

This is an infinite size LMI, but in fact only the constraints for $n = 0, 1, 3$ are active in the dual problem and so we need only consider $\psi \in \mathbb{R}^4$. The solution to the dual problem is readily obtained as $\nu_3 = 1.5449$ for $\gamma_1 = 9.4271, \gamma_2 = -3.8497, \delta = 7.8823$ and $\psi = [\,-4.5774 \;\; -6.28 \;\; 0 \;\; -1.319\,]^T$.

This means that the solution to the primal problem is only supported for $n = 0, 1, 3$ and we can easily solve to obtain $\hat{\phi}(\lambda) = 0.5807 + 0.7967\lambda + 0.1675\lambda^3$

In this problem we optimized an infinite size LMI, but once again the solution was FIR and hence computable exactly with a finite LMI. Note that the existence of a maximizing solution to the dual problem is guaranteed, and furthermore one can see that for any fixed $\gamma_1, \gamma_2 \in \mathbb{R}$ and $\psi \in \ell_2$ the constraints in (4.3) are decaying and will eventually become inactive. These two facts can be used to show that for this type of problem the solution is always FIR, and we state this as a theorem:

Theorem 4.4 *Consider a SISO (or square MIMO) ℓ_1 minimization problem, with no interpolation zeros on the unit circle, subject to an $\mathcal{H}_2$ constraint: $\|\phi\|_{\mathcal{H}_2} \leq \beta$ as described above. Then, provided there exists a stabilizing LTI controller satisfying $\|\phi\|_{\mathcal{H}_2} < \beta$, this problem has an FIR solution which is computable exactly with a finite LMI.*

This result was obtained independently in [12], and is the basis for the solution to the SISO mixed $\ell_1/\mathcal{H}_2$ problem presented there.

4.4 Robust $\mathcal{H}_2$ Control

We now consider the problem of minimizing the $\mathcal{H}_2$ norm of the closed loop map between some exogenous white noise inputs and some regulated outputs, while the ℓ_1 norm of the transfer function between inputs and outputs affected by model uncertainty is constrained to be below some level γ. The SISO case has been presented in [13]. This problem is not naturally posed as an LMI problem. However, through Theorem 3.1, it is possible to show that the problem has an equivalent primal-dual formulation in terms of LMI problems.

Such design will ensure good nominal rms-performance to white noise inputs and, at the same time, will guarantee stability robustness ([4, 2]) against LTV causal perturbations of induced ℓ_∞ norm less than $1/\gamma$. Formally, the problem we want to solve is given by:

$$\mu^o = \inf_{\Phi \in \ell_1^{n_z \times n_w}} \|\Pi_2 \Phi\|_2 \tag{10}$$

subject to: $\mathcal{A}\Phi = b$, and $\|\Pi_1 \Phi\|_1 \leq \gamma$
where Π_1 and Π_2 are projection operators from $\ell_1^{n_z \times n_w}$ to $\ell_1^{m \times n}$ for some $m \leq n_z$ and $n \leq n_w$.

Theorem 4.5 *Given Problem (10), assume that γ is strictly greater than ν^o, the optimal cost of the associated standard ℓ_1 problem:$\nu^0 = \inf_{\substack{\Phi \in \ell_1^{n_z \times n_w} \\ \mathcal{A}\Phi = b}} \|\Pi_1 \Phi\|_1$ then, the dual of Problem (10) with no duality gap is given by:*

$$\mu^o = \max \langle b, z^* \rangle - \gamma\eta \tag{11}$$

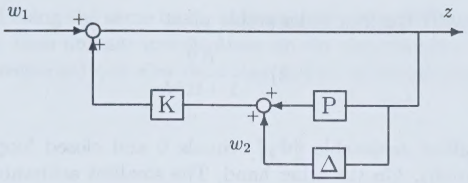


Figure 1: Filtering Problem Block Diagram

subject to: $\mathcal{A}^*z^* = \Pi_1^* w_1^* + \Pi_2^* w_2^*$, $\|w_2^*\|_2 \le 1$, $\|w_1^*\|_\infty \le \eta$, $\eta \in \mathbb{R}^+$, $z^* \in \ell_\infty$, *and* $w_1^* \in \ell_\infty^{m_1 \times n_1}$, $w_2^* \in \ell_2^{m_2 \times n_2}$.

For general multiblock cases we may need to look for finite-dimensional optimizations whose solutions approximate the optimal cost and/or the optimal solution.

In some special cases of multiblock problems, it is relatively easy to show that it is possible to approximate the optimal cost of the given dual problem by computing, via a finite-dimensional optimization, suboptimal dual solutions of finite support. Next, we present an example of a two-block problem for which the cost of feasible dual solutions with finite support approximates arbitrarily well the optimal cost. The approximation of the dual problem by finite support feasible solutions in the general case is under investigation.

4.4.1 Example

Consider the closed loop system described in Figure 4.4.1. Assuming the plant P is stable, we want to design a controller K, such that:

- the variance of the nominal error signal z ($\Delta = 0$) due to a white noise gaussian zero-mean input signal w_1 is minimized.
- the closed loop system is guaranteed to remain stable in presence of model uncertainty, represented as an additive perturbation Δ with $\|\Delta\|_{\ell_\infty - ind} < 1$.

In other words, we want to find a K such that $\|(1-PK)^{-1}\|_2$ is minimized and $\|K(1-PK)^{-1}\|_1 \le 1$. The closed loop map from (w_1, w_2) to z is given by: $\Phi = ((1-PK)^{-1}, K(1-PK)^{-1})$. The problem can be formally written as follows:

$$\mu^o = \inf_{\substack{k-stab \\ \|\Phi_2\|_1 \le 1}} \|\Phi_1\|_2$$

Corollary 4.1 *Assume that the constraints in the above problem is active. If a minimizing solution exists, then the optimal* Φ_2 *is* FIR. *Moreover,* μ^o *can be approximated with arbitrary accuracy by the cost of finite support dual feasible solutions.*

The dual problem assumes the following form:

$$\mu^o = \max_{\substack{\|z^*\|_2 \le 1 \\ \|\bar{P}^* z^*\|_\infty \le \eta \\ \eta \in \mathbb{R}^+,\ z^* \in \ell_\infty}} b^T z^* - \eta$$

For the above problem, it can also be shown that a minimizing optimal solution exists.

As an example, consider the first order stable plant

$$\hat{P}(\lambda) = \frac{0.2}{1+0.5\lambda}$$

Note that, the smallest achievable $\|\Phi_2\|_1$ equals 0 and closed loop that achieves it has $\|\Phi_1\|_2 = 1$ (no attenuation). On the other hand, The smallest achievable $\|\phi_1\|_2$ equals 0, but the closed loop that achieves it has $\|\Phi_2\|_1 = 7.5$. Thus, the stability robustness specification is not satisfied.

Computing cost of the mixed $\mathcal{H}_2/\ell_1$ problem from the dual problem we have that, for support of the dual solution $N \geq 5$, the sequence of sub-optimal dual costs practically converges to the value 0.8083.

For this problem it is shown in [16] that we can compute FIR suboptimal primal solutions whose costs approximate from above the optimal cost μ^o. This is done by by solving the following problem:

$$\begin{gathered} \overline{\mu}_N^2 = \min \quad \Phi_1^T\Phi_1 \\ \begin{bmatrix} I & -\bar{P} & \bar{P} \\ 0 & 1^T & 1^T \end{bmatrix} \begin{bmatrix} \Phi_1 \\ \Phi_2^+ \\ \Phi_2^- \end{bmatrix} \begin{matrix} = b \\ \leq \gamma \end{matrix} \\ \Phi_2^+ \geq 0,\ \Phi_2^- \geq 0 \\ \Phi_2^+(k) = 0,\ \Phi_2^-(k) = 0, \text{ for } k \geq N \end{gathered}$$

where 1^T is an infinite row vector with all elements equal to 1 and $\Phi_2 = \Phi_2^+ - \Phi_2^-$.

It is left to the reader to show that the above problem is equivalent to a finite LMI. It turns out that the optimal FIR solutions of length 10, 12, 14 have costs $\bar{\mu}_{10}^o = 0.8086$, $\bar{\mu}_{12}^o = 0.8084$ and $\bar{\mu}_{14}^o = 0.8083$. Thus, μ^o is equal to 0.8083 within a tolerance of 10^{-4}.

Actually, a closer look at the sequence of suboptimal FIR solutions allows us to compute exactly the solution to the problem. For each $N = 8, 10, 12, 14$, the optimal Φ_2 is such that $\Phi_2(k) = 0$ for $1 \leq k \leq N-2$. The $\Phi_2(0)$ and $\Phi_2(N-1)$ for each N are reported below:

N	$\Phi_2(0)$	$\Phi_2(N)$
8	−0.9922	−0.0078
10	−0.9981	−0.0019
12	−0.9995	−0.0005
14	−0.9999	−0.0001

Clearly, as N increases, the optimal Φ_2 converges to the sequence $\{-1 \quad 0 \quad 0 \quad \ldots\}$. Since $\Phi_2 = -Q$, the optimal Q for the mixed problem results equal to 1. The resulting optimal controller K is given by

$$K = -Q(1-PQ)^{-1} = -(1+P)^{-1}$$

The optimal Φ_1 is equal to $1-P$ with $\|\Phi_1\|_2 = 0.8083$.

5 Conclusions

In this paper we have presented a general framework for studying infinite dimensional LMI problems. The potential of this method lies in the generality of the LMI constraints that can be used to impose several performance specifications on the closed loop system. We have applied the duality results to solve the optimal ℓ_1 control problem with $\mathcal{H}_\infty$ constraints at a finite number of frequency points on the closed loop map, the mixed $\ell_1/\mathcal{H}_2$, and the robust $\mathcal{H}_2$, or mixed $\mathcal{H}_2/\ell_1$, control problem. In this last problem, we have allowed the different norm constraints to act on possibly different elements of the closed loop transfer function. Other problems can be studied using the same tools. In particular, since *LP* constraints are a special case of LMI constraints, time domain specifications on the response to some fixed input (e.g., templates for the step response) can also be added to the optimization problem, and this is detailed in [16].

References

[1] M.A. Dahleh, "BIBO Stability Robustness for Coprime Factor Perturbations," *IEEE Trans. Automat. Contr.*, Vol 37, pp. 352-355, 1992.

[2] M. Khammash and J. B. Pearson, "Performance Robustness of Discrete-Time Systems with Structured Uncertainty," *IEEE Trans. A-C* **36**, 398-412, 1991.

[3] J.S. McDonald and J.B. Pearson, "ℓ_1-Optimal Control of Multivariable Systems with Output Norm Constraints," *Automatica* **27**, 317-329, 1991.

[4] M.A. Dahleh and Y. Ohta, "A Necessary and Sufficient Condition for Robust BIBO Stability," *Systems & Control Letters,* **11**, 271-275, 1988.

[5] P. M. Young and M. A. Dahleh. "Infinite Dimensional Convex Optimization and Robust Control." submitted to *IEEE Trans. on Automatic Control.*

[6] J. C. Doyle, K. Glover, P. Khargonekar, and B. A. Francis, "State space solutions to $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems," *IEEE Transactions on Automatic Control*, vol. 34, pp. 831–847, 1989.

[7] M. A. Dahleh and J. B. Pearson, "ℓ_1 optimal feedback controllers for MIMO discrete-time systems," *IEEE Transactions on Automatic Control*, vol. 32, pp. 314–322, 1987.

[8] B. D. O. Anderson and J. B. Moore, *Linear Optimal Control*, Prentice Hall, New Jersey, 1971.

[9] J. Doyle, K. Zhou, K. Glover, and B. Bodenheimer "Mixed $\mathcal{H}_2$ and $\mathcal{H}_\infty$ performance objectives II: optimal control," *IEEE Transactions on Automatic Control*, vol. 39, pp. 1575–1587, 1994.

[10] D. S. Bernstein and W. M. Haddad, "LQG control with an $\mathcal{H}_\infty$ performance bound: a Riccati equation approach," *IEEE Transactions on Automatic Control*, vol. 34, pp. 293–305, 1989.

[11] M. Sznaier, "Mixed $\ell_1/\mathcal{H}_\infty$ controllers for MIMO discrete-time systems," in *Proceedings of the 33rd Conference on Decision and Control*, pp. 3187–3192, 1994.

[12] M. V. Salapaka, M. Dahleh, and P. G. Voulgaris, "Mixed objective control synthesis: optimal $\ell_1/\mathcal{H}_2$ control," in *American Control Conference*, pp. 1438–1442, 1995.

[13] P. Voulgaris, "Optimal $\mathcal{H}_2/\ell_1$ control: the SISO case," in *Proceedings of the 33rd Conference on Decision and Control*, pp. 3181–3186, 1994.

[14] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, SIAM, Philadelphia, 1994.

[15] M. A. Dahleh and I. J. Diaz-Bobillo, *Control of Uncertain Systems: A Linear Programming Approach*, Prentice-Hall, New Jersey, 1995.

[16] N. Elia and M. A. Dahleh, "Multiobjective controller design via infinite dimensional Linear Programming," submitted to *IEEE Transactions on Automatic Control.*

[17] N. Elia and M. A. Dahleh, "Controller design with multiple objectives" in *Proceedings of the American Control Conference*, pp. 1603–1607, 1994.

[18] D. G. Luenberger, *Optimization by Vector Space Methods*, John Wiley and Sons, New York, 1969.

On Reduced Order Mixed-Norm Control Synthesis: An Aircraft Terrain Following Example

David R. Jacques and D. Brett Ridgely
Air Force Institute of Technology/ENY
Wright Patterson AFB OH 45433-7765
email: dridgely@afit.af.mil

Abstract

This paper examines controller order trade-offs using a fixed-order, mixed-norm control synthesis technique for discrete-time linear systems. The method allows the control system designer to combine the H_2 , ℓ_1 and H_∞ norms of dissimilar transfer functions into a single constrained optimization problem. Any number or combination of constraints can be added to the problem, and the method constrains the norms directly without reliance on upper bounds to the norms. The method is applied to a terrain following problem involving a MIMO model of a fighter aircraft. An H_2/ℓ_1 problem is formulated and solved to provide altitude and flight path angle tracking in the presence of wind gusts and sensor noise. Finally, an H_∞ constraint on output complementary sensitivity is added to improve robust stability. Controllers are designed at both ninth and sixth order, with the latter representing a reduced order controller.

Introduction

Recently, there has been a great deal of interest in formulating and solving multiple objective optimal control problems capable of handling different classes of input and output signals. A typical multiobjective control problem would be to determine the trade-off between noise rejection (H_2) and some unstructured perturbation (H_∞) which embodies desired performance and robustness at either the input or output of the plant (or some combination). Concerns for control and rate saturation also make it desirable to be able to bound the magnitude of an output in response to a specific input (ℓ_∞) or a class of bounded magnitude inputs (ℓ_1). Because a real system is never limited to a single class of inputs or outputs, it is desirable to handle several classes of inputs and outputs within the framework of a single optimal control problem.

Ridgely and Walker[1] and Canfield, et. al.[2] have developed a numerical algorithm for solving the general H_2/H_∞ problem with output feedback. Their algorithm produces a suboptimal solution with the controller order fixed at some specifiable level. Their method has no requirements that the constraints be non-singular, and it allows for multiple H_∞ constraints.[3] This method was extended to discrete-time systems in Jacques and Ridgely,[4] which also included the addition of ℓ_1 constraints to the problem. As in the work by Ridgely and Canfield, the method allows any number of transfer function norms to be minimized or constrained directly, as opposed to limiting conservative upper bounds or approximations to the norms. Further, the constraint transfer functions can be singular.

In order to classify the nature of the solution, the theory assumes a unique optimal H_2 solution to the unconstrained problem. Therefore, the method basically assumes that the order of the compensator is chosen to be greater than or equal to the order of the H_2 part of the problem. A MATLAB[5] toolbox, MXTOOLS,[6] has been developed to solve these mixed-norm optimization problems. The method carries no claim of uniqueness or global optimality over all compensators; however, it has proven to be effective in balancing performance and robustness design constraints for a wide variety of problem tested to date.

This paper is an extension of the work in Jacques and Ridgely.[7] In Reference 7, a mixed $H_2/\ell_1/H_\infty$ design was performed on an A-4 aircraft to demonstrate robust terrain following. One of the attractive features of ℓ_1 optimization has always been that the ℓ_1 norm represents a bound on the maximum magnitude of the output for bounded magnitude inputs. In this regard, terrain following has often been mentioned as a logical application of ℓ_1 synthesis methods since it is highly desirable to maintain assurances on the maximum magnitude of altitude errors, not just the rms value. The design in Reference 7 produced a controller of order nine, which is the order of the H_2 subproblem. In this paper, a sixth order controller will be designed and compared to the ninth order controller. Necessary modifications to the theory and algorithms will be discussed.

Problem Setup

The system considered for the general mixed-norm control problem is shown in Figure 1. It contains three

---

[*] This paper is declared a work of the U. S. Government and is not subject to copyright protection in the United States. Cleared for public release. This work was sponsored in part by AFOSR.

sets of exogenous inputs and controlled outputs. In general, no relationship is assumed between r, d and w, or m, e and z. The input d is assumed to be a signal of unknown but bounded energy with $\|d\|_2 \leq 1$, and the input r is assumed to be a signal of unknown but bounded magnitude with $\|r\|_\infty \leq 1$. The input w is assumed to be the discrete-time equivalent of zero-mean white Gaussian noise of unit intensity. The state space of P is formed by wrapping the stable weights of an H_2 problem from w to z, the stable weights of an ℓ_1 problem from r to m and the stable weights of an H_∞ problem from d to e around the original plant. The order of the individual problems will in general be less than the order of P. An expanded state space realization of P can be written as

$$\begin{aligned} x_2(k+1) &= A_2 x_2(k) + B_w w(k) + B_2 u(k) \\ z(k) &= C_z x_2(k) + D_{zw} w(k) + D_{zu} u(k) \\ y_2(k) &= C_2 x_2(k) + D_{yw} w(k) + D_{yu} u(k) \end{aligned} \tag{1}$$

$$\begin{aligned} x_1(k+1) &= A_1 x_1(k) + B_r r(k) + B_1 u(k) \\ m(k) &= C_m x_1(k) + D_{mr} r(k) + D_{mu} u(k) \\ y_1(k) &= C_1 x_1(k) + D_{yr} r(k) + D_{yu} u(k) \end{aligned} \tag{2}$$

$$\begin{aligned} x_\infty(k+1) &= A_\infty x_\infty(k) + B_d d(k) + B_\infty u(k) \\ e(k) &= C_e x_\infty(k) + D_{ed} d(k) + D_{eu} u(k) \\ y_\infty(k) &= C_\infty x_\infty(k) + D_{yd} d(k) + D_{yu} u(k) \end{aligned} \tag{3}$$

where x_2, x_1 and x_∞ may have some or all states in common. The different y's represent the same measurement, but each has only the exogenous input associated with its subproblem. In the absence of disturbances, $y_2 = y_1 = y_\infty$; thus, $C_2 x_2 = C_1 x_1 = C_\infty x_\infty$. The true y is given by

$$y(k) = C_j x_j(k) + D_{yw} w(k) + D_{yr} r(k) + D_{yd} d(k) + D_{yu} u(k) \tag{4}$$

where j is 1, 2 or ∞, since the products are equal.

The objective of mixed $H_2/\ell_1/H_\infty$ control is to minimize the 2-norm of the closed-loop transfer function T_{zw}, while constraining the 1-norm of the transfer function T_{mr} and the ∞-norm of the transfer function T_{ed} to be less than some specified levels. A practical application might be to use the ∞-norm constraint to guarantee a certain level of stability robustness, while acceptable tracking error could be assured by enforcing a 1-norm constraint on the sensitivity, and the 2-norm could be used to optimize noise rejection while still meeting the other constraints. Mathematically, the problem can be stated as follows: For the system shown in Figure 1, find a stabilizing controller $K(z)$ of a specified order that achieves

$$\inf_{K admissible} \|T_{zw}\|_2 \tag{5}$$

subject to

$$\begin{aligned} \|T_{mr}\|_1 &\leq \nu \\ \|T_{ed}\|_\infty &\leq \gamma \end{aligned}$$

The following assumptions will be made:

i. (A_2, B_2) stabilizable, (C_2, A_2) detectable

ii. (A_1, B_1) stabilizable, (C_1, A_1) detectable

iii. (A_∞, B_∞) stabilizable, (C_∞, A_∞) detectable

iv. $D_{zu}^T D_{zu}$ full rank, $D_{yw} D_{yw}^T$ full rank

v. $\begin{bmatrix} A_2 - e^{j\theta} I & B_2 \\ C_z & D_{zu} \end{bmatrix}$ full column rank $\forall\ \theta \in [0, 2\pi)$

vi. $\begin{bmatrix} A_2 - e^{j\theta} I & B_w \\ C_2 & D_{yw} \end{bmatrix}$ full row rank $\forall\ \theta \in [0, 2\pi)$

vii. $D_{yu} = 0$

viii. At least one stabilizing controller of the chosen order exists

Conditions (i)-(iii) are necessary for the existence of stabilizing controllers of any order. If only stable weighting functions are used around the same core plant, the existence of a K which stabilizes the H_2 problem was shown to be necessary and sufficient for K stabilizing the H_∞ and ℓ_1 problems.[8] Conditions (iv)-(vi) ensure that the H_2 problem considered separately has a non-singular solution. A non-singular H_2 subproblem will not be necessary for the numerical solution, but it simplifies the design considerably. Numerically it will only be necessary that the overall mixed-norm problem be non-singular, so as to avoid solutions which require infinite control power or result in open-loop response characteristics. Condition (vii) is also not necessary, but a strictly causal P_{yu} or K is sufficient for well-posedness of the problem. A strictly causal P_{yu} is often a mild assumption when actuator dynamics are considered. Further, it frees us from making a similar assumption on the compensator, and it eases the notation required for this development. The relaxation of this condition is accomplished via a simple change of variables.[9] Define

$$y' = y - D_{yu} u \ \Rightarrow\ y = y' + D_{yu} u \tag{6}$$

The resulting plant now satisfies Condition (vii). The compensator that results from this is then shifted back to the original output – see Jacques and Ridgely[7] for the details. The resulting assumption is one of well-posedness.

Before discussing Condition (viii), we must make some definitions. As stated at the beginning of this section, we have assumed that we are trying to control some core plant, and have augmented that plant with interconnections and weightings – any dynamic weights are assumed stable. Now make the following definitions:

$$\alpha_o := \inf_{K stabilizing} \|T_{zw}\|_2 \tag{7}$$

$$\underline{\alpha} := \inf_{K admissible} \|T_{zw}\|_2 \tag{8}$$

$$K_{2_{opt}} := \text{the unique } K(z) \text{ that makes } \|T_{zw}\|_2 = \alpha_o \tag{9}$$

$$\underline{K}_2 := \arg\left\{ \inf_{K admissible} \|T_{zw}\|_2 \right\} \tag{10}$$

$$\nu_o := \inf_{K stabilizing} \|T_{mr}\|_1 \tag{11}$$

$$\underline{\nu} := \inf_{K admissible} \|T_{mr}\|_1 \tag{12}$$

$$\overline{\nu} := \|T_{mr}\|_1 \text{ when } K(z) = \underline{K}_2 \text{ or } K_{2_{opt}} \tag{13}$$

$$\gamma_o := \inf_{K stabilizing} \|T_{ed}\|_\infty \tag{14}$$

$$\begin{aligned}
\underline{\gamma} &:= \inf_{K admissible} \|T_{ed}\|_\infty & (15)\\
\overline{\gamma} &:= \|T_{ed}\|_\infty \text{ when } K(z) = \underline{K}_2 \text{ or } K_{2_{opt}} & (16)\\
n_g &:= \text{order of the core plant to be controlled} & (17)\\
n_2 &:= \text{order of the } H_2 \text{ subproblem} & (18)\\
n_1 &:= \text{order of the } \ell_1 \text{ subproblem} & (19)\\
n_\infty &:= \text{order of the } H_\infty \text{ subproblem} & (20)\\
n_c &:= \text{chosen order of } K & (21)
\end{aligned}$$

Admissible above means stabilizing *and* of a given controller order. If $n_c \geq \max\{n_1, n_2, n_\infty\}$, then there necessarily exists a stabilizing controller and $\{\nu_o, \alpha_o, \gamma_o\}$ are each achievable by a controller of the given order (not necessarily by the same controller, though). Next, assume $n_g < n_c < n_1$. Then a stabilizing controller necessarily exists, but in general ν_o is not achievable; the best we can do is some $\underline{\nu} > \nu_o$. The same arguement applies to the H_2 and H_∞ cases also. Finally, when $n_c < n_g$, Condition (viii) becomes necessary, as the chosen controller order may now be low enough that no stabilizing controllers exist. The question of what the minimum n_c is to satisfy Condition (viii) is an open research question.

The list of assumptions above represents a very nonrestrictive set. If (iv) - (vi) are relaxed and the shift in (6) is employed, they amount to requiring a problem which is stabilizable, detectable, well-posed, overall non-singular and such that at least one stabilizing controller exists at the chosen controller order.

The desired compensator K can be written in the form

$$\begin{aligned}
x_c(k+1) &= A_c x_c(k) + B_c y(k)\\
u(k) &= C_c x_c(k) + D_c y(k)
\end{aligned} \qquad (22)$$

where A_c, B_c, C_c and D_c are to be determined from the optimization problem. Note that for the continuous-time problem, D_c cannot be used as a design variable. This is because it is uniquely determined by

$$D_{zw} + D_{zu} D_c D_{yw} = 0 \qquad (23)$$

which is necessary in order for $\|T_{zw}\|_2$ to be finite.[1] This is not the case for discrete-time.

Using (22), the relevant closed-loop transfer functions can be written as

$$T_{zw} = \left[\begin{array}{c|c} \mathcal{A}_2 & \mathcal{B}_w \\ \hline \mathcal{C}_z & \mathcal{D}_{zw} \end{array}\right] \qquad (24)$$

$$T_{mr} = \left[\begin{array}{c|c} \mathcal{A}_1 & \mathcal{B}_r \\ \hline \mathcal{C}_m & \mathcal{D}_{mr} \end{array}\right] \qquad (25)$$

$$T_{ed} = \left[\begin{array}{c|c} \mathcal{A}_\infty & \mathcal{B}_d \\ \hline \mathcal{C}_e & \mathcal{D}_{ed} \end{array}\right] \qquad (26)$$

where

$$\begin{aligned}
\mathcal{A}_2 &= \begin{bmatrix} (A_2 + B_2 D_c C_2) & B_2 C_c \\ B_c C_2 & A_c \end{bmatrix}\\
\mathcal{A}_1 &= \begin{bmatrix} (A_1 + B_1 D_c C_1) & B_1 C_c \\ B_c C_1 & A_c \end{bmatrix}\\
\mathcal{A}_\infty &= \begin{bmatrix} (A_\infty + B_\infty D_c C_\infty) & B_\infty C_c \\ B_c C_\infty & A_c \end{bmatrix}
\end{aligned} \qquad (27)$$

$$\begin{aligned}
\mathcal{B}_w &= \begin{bmatrix} (B_w + B_2 D_c D_{yw}) \\ B_c D_{yw} \end{bmatrix}\\
\mathcal{B}_r &= \begin{bmatrix} (B_r + B_1 D_c D_{yr}) \\ B_c D_{yr} \end{bmatrix}\\
\mathcal{B}_d &= \begin{bmatrix} (B_d + B_\infty D_c D_{yd}) \\ B_c D_{yd} \end{bmatrix}
\end{aligned} \qquad (28)$$

$$\begin{aligned}
\mathcal{C}_z &= \begin{bmatrix} (C_z + D_{zu} D_c C_2) & D_{zu} C_c \end{bmatrix}\\
\mathcal{C}_m &= \begin{bmatrix} (C_m + D_{mu} D_c C_1) & D_{mu} C_c \end{bmatrix}\\
\mathcal{C}_e &= \begin{bmatrix} (C_e + D_{eu} D_c C_\infty) & D_{eu} C_c \end{bmatrix}
\end{aligned} \qquad (29)$$

$$\begin{aligned}
\mathcal{D}_{zw} &= \begin{bmatrix} D_{zu} D_c D_{yw} + D_{zw} \end{bmatrix}\\
\mathcal{D}_{mr} &= \begin{bmatrix} D_{mu} D_c D_{yr} + D_{mr} \end{bmatrix}\\
\mathcal{D}_{ed} &= \begin{bmatrix} D_{eu} D_c D_{yd} + D_{ed} \end{bmatrix}
\end{aligned} \qquad (30)$$

Numerical Approach

Canfield, Ridgely and Smith[2] used a sequential quadratic programming (SQP) algorithm to solve the continuous-time H_2/H_∞ problem. Applying their method to the current problem, the objective (f) and constraints (g) are as follows:

$$\begin{aligned}
f(\kappa) &:= \xi_2 \|T_{zw}\|_2^2 & (31)\\
g_1(\kappa) &:= \xi_1(\|T_{mr}\|_1 - \nu) & (32)\\
g_\infty(\kappa) &:= \xi_\infty(\|T_{ed}\|_\infty - \gamma) & (33)
\end{aligned}$$

where κ represents a vectorized compensator and the ξ's are scaling parameters. A modal form is assumed for the compensator in order to reduce the number of variables. Although this method does not allow repeated eigenvalues in the compensator, in practice it has shown itself to be sufficient. If it is deemed necessary to allow for repeated eigenvalues, a block-Jordan form or fully populated state-space could be used instead. Finally, because SQP searches over both feasible and infeasible solutions, a stability constraint and exterior penalty function were added to keep the algorithm from getting lost in an unstable region. The stability constraint is stated as

$$g_s(\kappa) = \xi_s(\max_i \{|\lambda_i(\mathcal{A}_2)|^2\} - 1) \qquad (34)$$

where $\lambda_i(\mathcal{A}_2)$ denotes the i^{th} eigenvalue of the closed-loop system, and $g_s(\kappa)$ is constrained to be less than zero. The penalty function added to the objective function is simply the square of the stability constraint, thus providing continuous derivatives at the stability boundary.

Further details on the numerical solution may be found in References 2 and 7. No changes to MXTOOLS are required for the reduced order problem. It is important to note that, in MXTOOLS, the order of the solution will always be fixed at the order of the initial compensator. Therefore, if a reduced order solution is desired, an initial stabilizing compensator with that order must be provided as the starting point. There are no restrictions as to how these initial stabilizing compensators may be found. Successful approaches include model order reduction of full-order compensators, and H_2, H_∞, or ℓ_1 synthesis of controllers for reduced order plants. Once a single stabilizing controller is found with the desired order, successful results from the mixed-norm problem will then provide the necessary compensators for starting the algorithm with new constraint levels. This is perhaps the most difficult aspect of using the method for reduced order mixed controllers – work is currently underway to simplify this process.

MIMO Aircraft Terrain Following Example

Problem Overview

The example used will be a longitudinal controller design for a fighter aircraft in a terrain following mode. The aircraft is an A-4 operating at sea level, Mach 0.85.[10] There are five states in the basic plant: vertical velocity in the aircraft axes (w, meters/sec), altitude perturbation (h, meters), pitch rate (q, radians/sec), pitch angle (θ, radians) and forward velocity perturbation in the aircraft axes (u, meters/sec). The control inputs are elevator deflection (δ_e, radians) and throttle (δ_T). A throttle control was specifically included in this example to help maintain forward velocity during a commanded change in altitude. The throttle repsonse was assumed to be more sluggish than that of the elevator, and this makes it less attractive for controlling the states which vary quickly. However, natural forward velocity variations are mainly due to the slowly varying phugoid mode, and the throttle provides excellent control authority over this state. Measurements are assumed available from a combination of air data and some type of inertial measurement unit, and measurements consist of altitude rate, altitude, normal acceleration felt by the pilot (n_z), pitch angle, and forward velocity perturbation. Commanded inputs will be a combination of flight path angle (or altitude rate) and altitude. Controlled outputs will be various combinations of the measurements. The goal is to enable the aircraft to follow flight path angle and altitude commands quickly and accurately, maintain low overshoot and reasonable settling times, keep control usage and pilot g-levels within reasonable bounds, and have the steady state flight be relatively insensitive to wind gusts and measurement noise. The continuous-time plant is given by

$$\begin{Bmatrix} \dot{w} \\ \dot{h} \\ \dot{q} \\ \dot{\theta} \\ \dot{u} \end{Bmatrix} = \begin{bmatrix} -2.23 & 0 & 289.13 & 0 & -0.1134 \\ -1 & 0 & 0 & 289.13 & 0 \\ -0.15513 & 0 & -4.1708 & 0 & 0.0041218 \\ 0 & 0 & 1 & 0 & 0 \\ -0.0368 & 0 & 0 & -9.8066 & -0.0308 \end{bmatrix} \begin{Bmatrix} w \\ h \\ q \\ \theta \\ u \end{Bmatrix} + \begin{bmatrix} -57.302 & 0 \\ 0 & 0 \\ -63.754 & .000243 \\ 0 & 0 \\ 0 & .002344 \end{bmatrix} \begin{Bmatrix} \delta_e \\ \delta_T \end{Bmatrix} \quad (35)$$

$$\begin{Bmatrix} \dot{h} \\ h \\ n_z \\ \theta \\ u \end{Bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 289.13 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ -0.17918 & 0 & 30.780 & 0 & -.01284 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} w \\ h \\ q \\ \theta \\ u \end{Bmatrix} \quad (36)$$

Actuator dynamics are assumed for both the elevator and the throttle, and are modelled as follows:

$$\begin{aligned} \delta_e(s) &= \frac{20}{s+20}\delta_{e_c}(s) \\ \delta_T(s) &= \frac{5}{s+5}\delta_{T_c}(s) \end{aligned} \quad (37)$$

As mentioned previously, the throttle was assumed to be more sluggish than the elevator in order to discourage its use for controlling the faster varying modes of the system.

The basic plant includes the short period mode ($\omega_{sp} = 7.35$ rad/sec), phugoid mode ($\omega_p = 0.0696$ rad/sec), and an additional pole at the origin resulting from the altitude state. The wide separation in frequency between the modes presents a challenging problem; a sample rate fast enough to accommodate the short period and actuator modes will result in high truncation levels in order to capture the phugoid and integrator state. A sample rate of 20 Hz was chosen based on the actuator modes ($6.3 \times \omega_{\delta_e}$), and a zero-order hold was used for discretization of the transfer functions. Simulations of the various closed-loop systems used a continuous-time plant and discrete-time compensators. Further, in order to provide a more realistic reference command (as opposed to a discrete step input) with which to evaluate the controllers, a command generator was used in the closed-loop simulations (see Figure 2). The command generator provides a commanded increase in altitude of 120 meters which takes place over 3 seconds. The command filter with $\omega_c = 5$ rad/sec was chosen to model either pilot lag or command smoothing on the part of an automatic terrain following mode. As a pilot model, the command filter represents a "fast" pilot, thus presenting a more challenging design problem.

H_2 Subproblem

The H_2 subproblem is based on a steady state Linear Quadratic Gaussian (LQG) design. A wind disturbance, which enters the plant as an angle of attack disturbance, is modelled as the output of a first-order shaping filter driven by zero-mean white Gaussian noise, and is included in both the design model and the simulation model. Intensity levels and time constants are chosen for sea level flight in cumulus clouds as used for the control designs on the NASA F-8C.[11] The wind gust state and the input vector describing how it enters the plant are as follows:

$$\dot{w}_g = -9.486\ w_g + w \quad (38)$$

$$B_{w_g} = \begin{bmatrix} -7.713 \times 10^{-3} \\ 3.459 \times 10^{-3} \\ -5.365 \times 10^{-4} \\ 0 \\ -9.807 \end{bmatrix} \quad (39)$$

Controlled outputs consist of the measurements with weightings of 10, 1000, 1000, 10 and 1 on $\dot{h}$, h, n_z, θ and u, respectively. To enable non-zero set point tracking, a pseudo-integral state on altitude error was added to the plant as follows:

$$i_h(s) = \frac{1000}{s+0.001}\ (h(s)\ -\ h_c(s)) \quad (40)$$

and a weight of 0.1 was used for the output of the pseudo-integral state. Control weights of 0.1 and 1 were used for

δ_e and δ_T, respectively. All of the weights were chosen subjectively and iteratively based on closed-loop simulations, with a goal of maintaining steady forward velocity during a commanded change in altitude, maintaining g-levels below 4 g's for simulated commands, and accurately tracking altitude and altitude rate commands. The most difficult of these design goals was the maintainance of g-levels, but this was partially due to the "fast" pilot model being used for evaluation. Naturally, for any given controller, a slower command will result in a lower g-level and a more sluggish response. The higher weight on δ_T as opposed to that on δ_e is again meant to discourage the use of thrust over elevator commands. Thrust is primarily intended to maintain forward velocity.

The measurements are assumed to be corrupted by zero-mean, white Gaussian noises of strength 10^{-5} rad^2–sec. It was decided to allow high overshoot and large pitch variations in the H_2 portion of the problem in hopes that it could be significantly reduced using an ℓ_1 constraint. Overall, the H_2 subproblem provides quick response to altitude and altitude rate errors, and is generally unaffected by wind gusts and measurement noise. The noise levels used for the simulation were generally higher than those used to synthesize the controller. This was done in order to highlight differences between the controllers.

The H_2 plant for this example is ninth order, which will also be the order of the H_2 optimal compensator. With a ninth order compensator, the total number of design variables for this example will be 90, of which 45 will be in the compensator B_c matrix. For the sixth order compensator, there will be 63 design variables, almost one-third less. Both of these represent very challanging nonlinear optimization problems.

ℓ_1 Subproblem

The ℓ_1 subproblem was used to reduce the maximum magnitude of the altitude error, thereby reducing the large overshoot resulting from the H_2 subproblem. The bounded magnitude inputs were taken as the wind gust and commanded altitude and altitude rate (or flight path angle). We wish to be able to reject the wind gust while tracking the altitude and altitude rate commands. A more general sensitivity constraint could have been used as an alternative, but the H_2 subproblem provided adequate state regulation, and we were exclusively interested in constraining altitude error. The controlled altitude error was band limited in order to reduce the susceptibility to measurement noise,

$$h_o(s) = \frac{4}{s+4}\,\left(h(s) - h_c(s)\right) \qquad (41)$$

and a weight of 1 was used. The value of the weight for the ℓ_1 constraint is unimportant for this problem because it is a single output constraint. The bandwidth of the frequency weight was based on the pilot model, and could have been set to 5 rad/sec to coincide with the command filter used for simulation. However, since the ℓ_1 gradient algorithm switches to finite difference methods for the case of repeated roots, the bandwidth was dropped to 4 rad/sec so as not to coincide with the open-loop pole from the throttle dynamics. This still represents relatively fast pilot commands for the design model, and it allows the ℓ_1 gradient algorithm to take advantage of the analytic formulations. The order of the ℓ_1 subproblem is eight.

H_∞ Subproblem

The H_∞ subproblem was used to improve the robustness of the design. Output complimentary sensitivity was chosen for T_{ed}, reflecting an output multiplicative uncertainty. The weight was chosen as unity, since the typical high frequency weight for unmodeled dynamics would take effect beyond the Nyquist sampling frequency for this system. Thus, the purpose of minimizing the ∞-norm of T_{ed} here was to improve stability margins by reducing the peak value of complimentary sensitivity. The H_∞ subproblem was order seven.

H_2 Results

The ninth order H_2 design was performed using standard MATLAB functions, and produced an optimal 2-norm of $\alpha_o = 0.1584$. To find a sixth order H_2 "optimal" controller, MXTOOLS was used. First, note that reduced order H_2 controllers are not necessarily unique. To find a (possibly local) minimum, first a sixth order stabilizing controller was found. For this example, this was done by stripping the two actuator states and the wind state out of the H_2 subproblem and performing H_2 optimization on the resulting sixth order plant. This controller was then checked on the ninth order system and was found to be stabilizing. This controller was then given as a starting controller for a mixed H_2/ℓ_1 run with the ℓ_1 constraint set to a very high value (to make the constraint inactive). The code was run several times from various intermediate controllers to try to "shake out" any local minima. This does not provide any global guarantees, but good results were obtained. The minimum value of the 2-norm was found to be $\underline{\alpha} = 0.15877$ for a sixth order controller. Figure 3 shows the altitide response for the commanded 40 m/sec climb held for 3 sec, for both the ninth and sixth order controllers. The large overshoot is especially troublesome, because it represents increased vulnerability of the aircraft. It also represents a potentially hazardous situation for the pilot if a decrease in altitude were commanded in low level flight. Interestingly the overshoot is reduced for the sixth order controller.

An additional drawback of the large overshoot associated with the H_2 solutions is evident in the pitch angle responses shown in Figure 4. Neglecting angle of attack a 40 m/sec climb represents an 8 deg pitch angle. Note that the ninth order H_2 solution has a pitch response which peaks at close to 15 deg, and the sixth order peak is around 13 deg. Also, a large negative pitch angle (-

Table 1 Mixed H_2/ℓ_1 Control Results

Case #	$\|\|T_{mr}\|\|_1$	(9th) $\|\|T_{zw}\|\|_2$	(6th) $\|\|T_{zw}\|\|_2$
1 (H_2 optimal)	3.086 ($\overline{\nu}_9$)	0.15840	—
2	2.800	0.15840	—
(H_2 red opt)	2.771 ($\overline{\nu}_6$)	—	0.15877
3	2.600	0.15841	0.15876
4	2.400	0.15844	0.15877
5	2.200	0.15851	0.15883
6	2.000	0.15861	0.15900
7	1.800	0.15907	0.15983
8	1.600	0.16164	0.16175

deg for ninth, -5 deg for sixth) results from the rapid decrease in altitude as the compensator attempts to damp out the large error associated with the overshoot.

Finally, the g-loading on the pilot for both controllers is shown in Figure 5. The ninth order controller definitely exceeds the 4-g limit, and the sixth order just barely exceeds it.

H_2/ℓ_1 Results

Table 1 shows norm values for several of the H_2/ℓ_1 compensators, for both ninth and sixth order. Figure 6 shows the Pareto-optimal curves for each order. Note that the sixth order curve is smoother – this represents the decreased numerical difficulties resulting from a problem with fewer design variables. As expected, the lower order curve is always above the higher order one, indicating the trade-off in performance versus controller order. Figure 7 shows the altitude response for the ninth order controllers numbered in Table 1. As shown, the ℓ_1 constraint can significantly reduce the overshoot without sacrificing rise time and overall speed of response. Figure 8 shows a comparison of the altitude response for the ninth and sixth order controllers represented by Case 6 from Table 1. The rationale behind choosing this case will become clear in the next section. Note that both responses are similar, but the sixth order may actually be better as it gets closer to the final value more quickly. The settling time for the H_2/ℓ_1 systems could be improved by using several methods. In addition to the pseudo-integral state added to the H_2 subproblem, a similar weight on altitude error could have been added to the ℓ_1 subproblem. A second method, which would have required only minor modification to the algorithms, would be to add a decaying exponential (time-domain) weight on the ℓ_1 transfer function.

Figure 9 shows the pitch angle response for the ninth order H_2/ℓ_1 systems from Table 1. Note that the addition of the ℓ_1 constraint quickly reduces the overshoot that is beyond the desired 8 deg as well as the undershoot associated with leveling off. These are clearly desirable features of the H_2/ℓ_1 systems. Figure 10 shows the pitch response for Case 6 for the full and reduced order controllers. This shows very little difference between the two, with the sixth order slightly worse due to the slight decrease in overshoot in the altitude response.

One of the drawbacks of the H_2/ℓ_1 systems is increased g-loading on the pilot. As the ℓ_1 constraint is lowered, the maximum g-level is increased. However, for the ninth order controllers, as long as $\nu > 2$, the 4-g limit is met. Figure 11 shows a comparison of the Case 6 acceleration responses for the full and reduced order controllers, with the sixth order being slightly worse. Finally, the commanded elevator usage actually decreases for the H_2/ℓ_1 systems versus the H_2 optimal controllers. The peak elevator usage for the H_2 systems is about 5 deg, but only 3 deg for Case 6. As the ℓ_1 norm is reduced further, the elevator rates become larger, but still reasonable (about 20 deg/sec for Case 8).

$H_2/\ell_1/H_\infty$ Results

For the H_2/ℓ_1 cases discussed above, no attempt was made to monitor or improve robust stability levels for the closed-loop system; these concerns were temporarily set aside for the H_∞ subproblem. Using the ninth order controllers in Cases 1-8 from the previous section and examining the resulting $||T_{ed}||_\infty$, it was seen that the ∞-norm and 1-norm constraints are not competing until $||T_{mr}||_1$ is pushed well below 2.0, after which the value of $||T_{ed}||_\infty$ increases significantly. Based on these curves, and the time responses shown previously, a 1-norm constraint level of $\nu = 2.0$ was chosen as the desired level of tracking. Holding this 1-norm constraint level, $||T_{ed}||_\infty$ was then reduced from a starting value of $\gamma = 2.14$ for the ninth order case and a starting point of 4.4683 for the sixth order case using $H_2/\ell_1/H_\infty$ optimization. The resulting Pareto-optimal curves are shown in Figure 12. The effect of reducing the singular value peaks of T_{ed} can clearly be seen in Figure 13 for the ninth order case, and the appearance of multiple active peaks for the lower γ levels is evident.

Table 2 shows the values of the norms and the corresponding independent phase margins of the $H_2/\ell_1/H_\infty$ designs. The gain margins are omitted for brevity. Significant improvements can be made in the margins with only slight increases in $||T_{zw}||_2$. Note that the Case 12 systems for both orders have the same 1-norm and ∞-norm and yield almost identical 2-norms. It should therefore be little surprise that the time responses are almost identical, as shown in Figures 14, 15 and 16. An unexpected bonus of improving the robust stability for this particular problem is that g-levels and control usage actually decreased. The target level of 4-g's is met, and maximum elevator deflections are very low ($2 - 3$ deg) for the simulated commands.

Table 2 Mixed $H_2/\ell_1/H_\infty$ Control Results

Case #	$\|\|T_{ed}\|\|_\infty$	(9th) $\|\|T_{zw}\|\|_2$	(9th) $IPM(deg)$	(6th) $\|\|T_{zw}\|\|_2$	(6th) $IPM(deg)$
6 (6th)	4.4683	—	—	0.15900	16.2
6 (9th)	2.1400	0.15861	27.1	—	—
9	2.0000	0.15864	29.0	0.15916	29.0
10	1.8000	0.15879	32.3	0.15935	32.2
11	1.6000	0.15906	36.5	0.15963	36.4
12	1.4000	0.16507	42.0	0.16033	41.9

Conclusions

As demonstrated by the terrain following example, the proposed method is capable of combining the H_2, ℓ_1 and H_∞ norms of different transfer functions into a single, constrained optimization problem. The method has great practical potential because it yields compensators which are of a specifiable fixed order. This was clearly demonstrated in this paper. This feature of the method eliminates the need for further model reduction. Performing model order reduction on the Case 12 ninth order controller to reduce it to sixth order resulted in a very poor controller, in terms of both performance and robustness. Although the method makes no claims of convexity, global optimality or convergence, it has proven to be effective with a variety of problems. Tradeoff curves showing objective vs. controller order for given constraint levels can be generated by exercising the algorithm at various controller orders. In particular, the sixth order mixed $H_2/\ell_1/H_\infty$ controller designed in this paper was just as good as the ninth order one.

References

[1] D. B. Ridgely and D. Walker. The general mixed H_2/H_∞ control problem. Submitted to *Automatica*, 1995.

[2] R.A. Canfield, D. B. Ridgely, and L. Smith. Numerical solution to the fixed-order, general mixed H_2/H_∞ optimal control problem. Submitted to *Automatica*, 1995.

[3] J. Ullauri, D. E. Walker, and D. B. Ridgely. Reduced order mixed H_2/H_∞ optimization with multiple H_∞ constraints. In *AIAA Guidance, Navigation, and Control Conference*, pages 1051–1060, Scottsdale, AZ, August 1994.

[4] D. R. Jacques and D. B. Ridgely. A fixed-order, mixed-norm control synthesis method for discrete linear systems. In *Proceedings of the 1995 American Control Conference*, pages 1936–1940, Seattle, WA, June 1995.

[5] MATLAB: High performance numeric computation and visualization software. The Math Works, Inc., Natick MA, 1994.

[6] D. R. Jacques, D. B. Ridgely, R. A. Canfield, and M. A. Spillman. A MATLAB toolbox for fixed-order, mixed-norm control synthesis. In *Proceedings of the 1995 IEEE Conference on Control Applications*, Albany, NY, September 1995.

[7] D. R. Jacques and D. B. Ridgely. Discrete-time, mixed-norm control synthesis applied to aircraft terrain following. In *Proceedings of the 1995 AIAA Guidance, Navigation and Control Conference*, pages 601–611, Baltimore, MD, August 1995.

[8] D. E. Walker. *H_2 Optimal Control with H_∞, μ, and L_1 Constraints*. PhD thesis, Air Force Institute of Technology, WPAFB, OH, June 1994.

[9] M. G. Safonov, D. J. N. Limebeer, and R. Y. Chiang. Simplifying the H_∞ theory via loop-shifting, matrix-pencils and descriptor concepts. *International Journal o Control*, 50(6):2467–2488, 1989.

[10] D. McRuer, I. Ashkenas, and D. Graham. *Aircraf Dynamics and Automatic Control*. Princeton University Press, Princeton, NJ, 1973.

[11] M. Athans et al. Stochastic control of the F-8C aircraft using a multiple model adaptive control (MMAC) method - Part 1: Equilibrium flight. *IEEE Trans. on Auto. Control*, AC-22(5):768–780, October 1977.

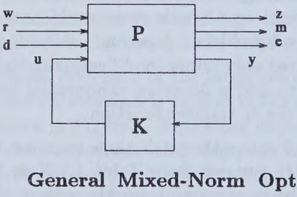


Fig. 1 General Mixed-Norm Optimizatio Problem

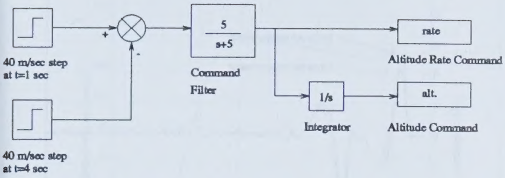

40 m/sec step
at t=1 sec
+
-
5
s+5
Command
Filter
rate
Altitude Rate Command
1/s
Integrator
alt.
Altitude Command
40 m/sec step
at t=4 sec

Fig. 2 Command Generator Used for Simulation

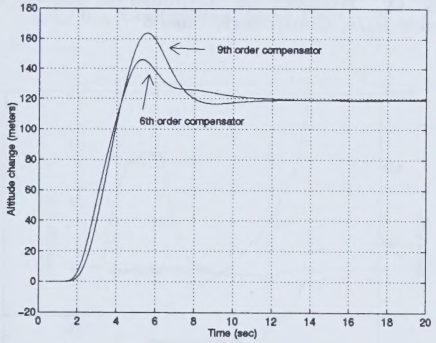


Fig. 3 Altitude Response for 9th vs. 6th Order H_2 Controllers

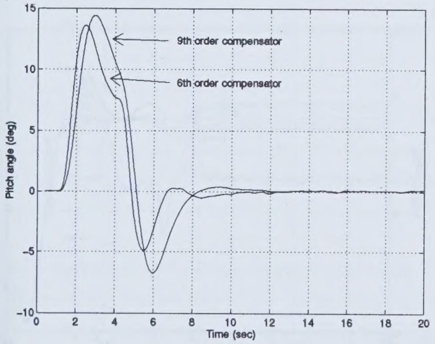


Fig. 4 Pitch Response for 9th vs. 6th Order H_2 Controllers

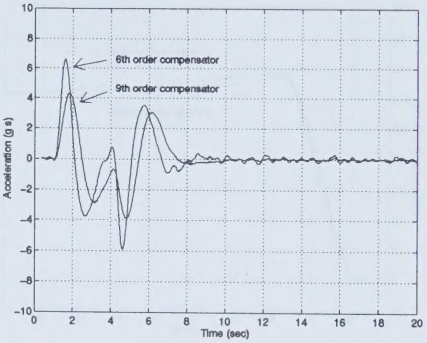


Fig. 5 Acceleration Response for 9th vs. 6th Order H_2 Controllers

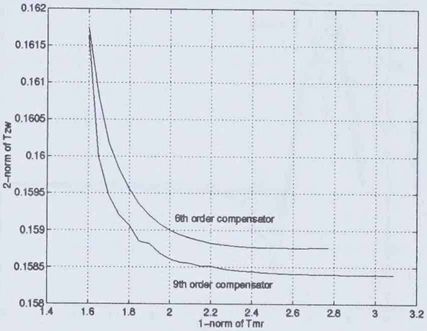


Fig. 6 Pareto-optimal H_2/ℓ_1 Curves, 9th vs. 6th Order

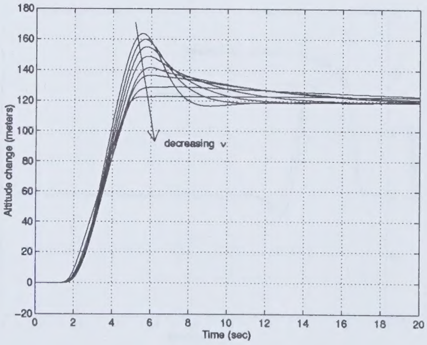


Fig. 7 Altitude Response for 9th Order H_2/ℓ_1 Designs, Cases 1-8

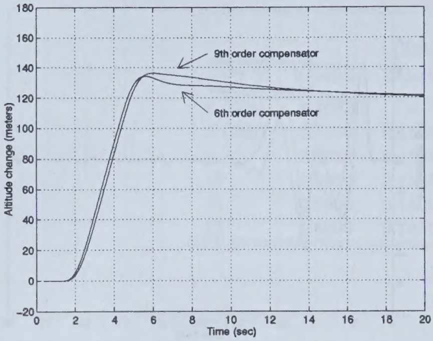


Fig. 8 Altitude Response for 9th vs. 6th Order H_2/ℓ_1 Controllers, Case 6

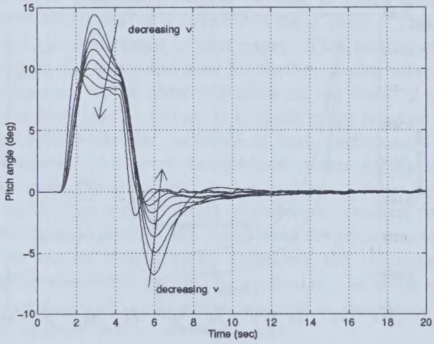


Fig. 9 Pitch Response for 9th Order H_2/ℓ_1 Designs, Cases 1-8

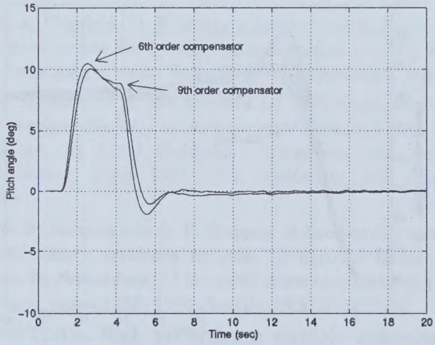


Fig. 10 Pitch Response for 9th vs. 6th Order H_2/ℓ_1 Controllers, Case 6

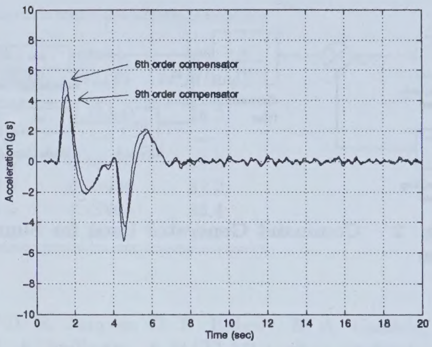


Fig. 11 Acceleration Response for 9th vs. 6th Order H_2/ℓ_1 Controllers, Case 6

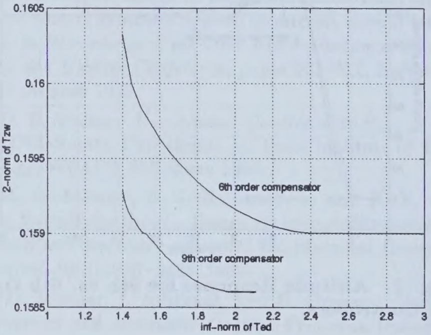


Fig. 12 Pareto-optimal $H_2/\ell_1/H_\infty$ Curves, $\nu =$ 2.0, 9th vs. 6th Order

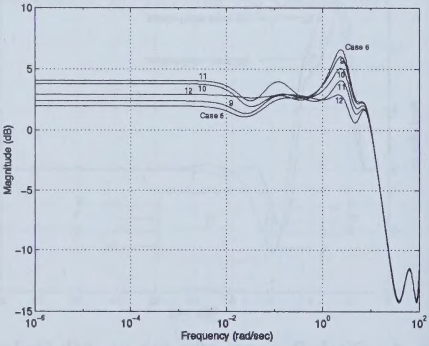


Fig. 13 Output Complementary Sensitivity for $H_2/\ell_1/H_\infty$ Designs, $\nu = 2.0$

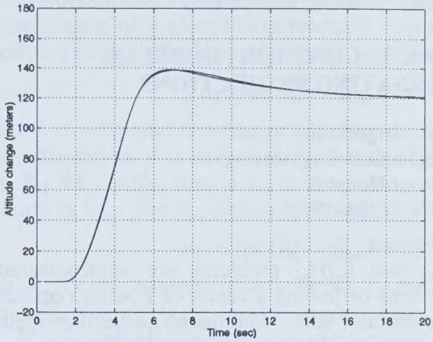


Fig. 14 Altitude Response for 9th vs. 6th Order $H_2/\ell_1/H_\infty$ Controllers, Case 12

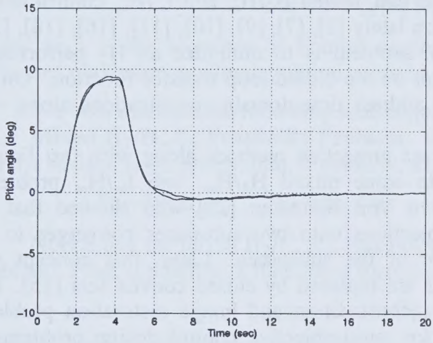


Fig. 15 Pitch Response for 9th vs. 6th Order $H_2/\ell_1/H_\infty$ Controllers, Case 12

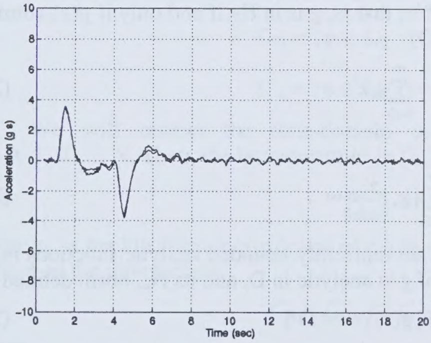


Fig. 16 Acceleration Response for 9th vs. 6th Order $H_2/\ell_1/H_\infty$ Controllers, Case 12

SOLUTION OF MIXED $H_2/H_\infty/l_\infty$ CONTROL PROBLEMS USING FFTs AND ALTERNATING PROJECTIONS

Karolos M. Grigoriadis
Department of Mechanical Engineering
University of Houston
Houston, TX 77204-4792

Abstract

In this paper, multiobjective H_2/H_∞ and l_∞/H_∞ problems are approximated in frequency domain as convex feasibility problems of finding a vector of Fourier coefficients in the intersection of a family of convex constraint sets. Alternating projection methods along with fast Fourier transform techniques are proposed for numerical solution.

1. Introduction

Practical control design problems must take into consideration both performance specifications and model uncertainty. To this end, mixed H_2/H_∞ and l_∞/H_∞ control design problems have received considerable attention lately [2], [7], [9], [10], [11], [16], [18], [21], [22]. The goal of a mixed H_2/H_∞ control problem is to minimize an H_2 performance criterion, subject to a specified H_∞ constraint on the closed-loop transfer function. On the other hand, mixed l_∞/H_∞ control problems address time-domain specifications along with H_∞ constraints.

In this paper, we use alternating convex projection methods along with fast Fourier transform techniques to obtain solutions to some mixed H_2/H_∞ and l_∞/H_∞ problems. Alternating projection methods date back to Von Neumann [23] who showed that the sequence of the alternating orthogonal projections onto two subspaces converges to the orthogonal projection onto the intersection of the subspaces. Later, this concept was generalized to the case where the subspaces are replaced by closed convex sets [15]. This result had extensive applications in signal reconstruction and image restoration problems [24]. Recently, these methods were used for multi-objective control design problems in state-space [13], [14] and solution of mixed H_2/H_∞ problems in the frequency domain [11].

2. Basic Definitions and Notation

Recall that H_2 is the Hardy space of all analytic functions g in the open unit disc D whose Taylor coefficients are square summable, that is, g is in H_2 if and only if g(z) admits a power series expansion of the form (for z in D)

$$g(z) = \sum_{i=0}^{\infty} g_i z^i \tag{2.1}$$

where the H_2 norm of g satisfies

$$\|g\|_2^2 = \sum_{i=0}^{\infty} |g_i|^2 < \infty \; . \tag{2.2}$$

On the other hand, H_∞ is the Hardy space of all uniformly bounded analytic functions in the open unit disc, that is, g is in H_∞ if and only if g is analytic in D, and its H_∞ norm defined by

$$\|g\|_\infty = \sup\{\, |g(z)| \; : z \in D \,\} \tag{2.3}$$

is finite. Obviously, H_∞ is a subset of H_2. The Banach space l_∞ is the vector space of bounded real sequences $\{e_k\}_{k=0}^{\infty}$ equipped with the norm

$$\|\{e_k\}\|_{l_\infty} = \sup\{\, |e_k| \; : k = 0,\ 1,\ 2, \ldots \} \tag{2.4}$$

Utilizing the Youla-Kucera parametrization of all stabilizing controllers, H_2, H_∞ and l_∞ constraints of a closed-loop feedback system can be formulated in terms of a rational function q in H_∞. The H_2 optimization problem is

$$d_2 = \inf \{ \|f - q\|_2 \ : \ q \in H_\infty \} = \inf \|f - H_\infty\|_2 \tag{2.5}$$

where f is a rational function in L_∞. Notice that d_2 is the distance from f to H_∞ in the L_2 norm. Moreover, it is easy to show that $d_2 = \|P_- f\|_2$, where P_- is the orthogonal projection onto $K_2 \overset{\Delta}{=} L_2 \ominus H_2$, that is K_2 is the orthogonal complement of H_2 in L_2. Let g be any function in L_∞. The H_∞ Nehari optimization problem is

$$d_\infty = \inf \{ \|g - q\|_\infty \ : \ q \in H_\infty \} = \inf \|g - H_\infty\|_\infty \, . \tag{2.6}$$

Here d_∞ is the distance from g to H_∞ in the L_∞ norm. By the famous Nehari theorem, $d_\infty = \|\Gamma\|$, where Γ is the Hankel operator from H_2 into K_2 defined by $\Gamma h = P_- gh$ for all h in H_2. If g is rational, then one can use the state-space technique by Glover [13] to compute d_∞, that is, d_∞ is the largest eigenvalue of PQ where P is the controllability Gramian and Q is the observability Gramian of a minimal realization for g. Given two rational functions r and t in L_∞, the l_∞ optimization problem can be written as [7]

$$d_t = \inf \{ \|\{e_k\}\|_{l_\infty} \ : \ q \in H_\infty \}$$

where $\{e_k\}$ is the sequence of Fourier coefficients of the function $r - tq$. This problem addresses maximum amplitude minimization to a given fixed input

We now consider the following multiobjective feasibility problem.

Mixed $H_2/H_\infty/l_\infty$ Feasibility Problem: Given the rational functions f, g, r and t in L_∞ and the scalars γ_2, γ_∞ and γ_t, find, if possible, a q in H_∞ satisfying

$$\|f - q\|_2 \le \gamma_2 \, , \quad \|g - q\|_\infty \le \gamma_\infty \quad \text{and} \quad \|r - tq\|_{l_\infty} \le \gamma_t \quad (q \in H_\infty) \, . \tag{2.7}$$

Notice that the mixed $H_2/H_\infty/l_\infty$ feasibility problem (2.7) is equivalent to the following problem: Find, if possible, a function q such that

$$q \in C_1 \cap C_2 \cap C_3 \cap C_4 \tag{2.8}$$

where C_1 and C_2 are the convex sets in L_2 defined by

$$C_1 = H_2 \tag{2.9}$$

$$C_2 = \{ \phi \in L_2 \ : \ \|g - \phi\|_\infty \le \gamma_\infty \} \tag{2.10}$$

$$C_3 = \{ \phi \in L_2 \ : \ \|f - \phi\|_2 \le \gamma_2 \} \tag{2.11}$$

$$C_4 = \{ \phi \in L_2 \ : \ \|\{e_k\}\|_{l_\infty} \le \gamma_t \} \, . \tag{2.12}$$

We will restrict the time-domain constraints to hold over a finite horizon $k = 0, 1, 2, ..., N$. Then, the last constraint in (2.7) is equivalent to

$$\max_{0 \le k \le N} \ |e_k| \le \gamma_t$$

where e_k is the kth Fourier coefficient of $r - t\phi$. Hence, the constraint set C_4 in (2.12) can be written as the intersection

$$C_4 = C_{41} \cap C_{42} \cap ... \cap C_{4N} \tag{2.14}$$

where

$$C_{4k} = \{ \phi \in L_2 \ : \ |e_k| \le \gamma_t \} \, , \quad k = 1, 2, ..., N \, . \tag{2.15}$$

The multiobjective feasibility problem (2.7) is equivalent to finding, if exists, a q in H_∞ satisfying

$$q \in C_1 \cap C_2 \cap C_3 \cap C_{41} \cap \ldots \cap C_{4N} \tag{2.16}$$

Throughout, P_i is the projection from L_2 onto the convex set C_i, for $i = 1, 2, 3, 41, \ldots, 4N$. Recall that by the projection theorem, P_i is the (possibly nonlinear) operator defined by $P_i\alpha = \alpha^*$ where α is in L^2 and α^* is the unique element in C_i solving the following n problem

$$\|\alpha - \alpha^*\|_2 = \inf \{ \|\alpha - \beta\|_2 \ : \beta \in C_i \} . \tag{2.17}$$

Moreover, $\alpha^* = P_i\alpha$ is the unique element in C_i satisfying the following "maximum principle" condition

$$\operatorname{Re} (\alpha - \alpha^*, \alpha^* - \beta) \geq 0 \qquad (\text{for all } \beta \in C_i) . \tag{2.18}$$

Our procedure in solving the feasibility problem (2.16) is to use alternating projection methods on the convex sets described by C_1, C_2 and C_3. For a function α in L^2, let $\{\alpha_n\}_{n=-\infty}^{\infty}$ be the coefficients of the Fourier series expansion of α, that is

$$\alpha(z) = \sum_{n=-\infty}^{\infty} \alpha_n z^n \qquad (\text{for } z = e^{it}) . \tag{2.19}$$

The projection $P_1 = P_+$ onto $C_1 = H_2$ is the linear operator on L^2 defined by

$$P_1 \alpha = \sum_{n=0}^{\infty} \alpha_n z^n \qquad (\alpha \in L_2) . \tag{2.20}$$

It is easy to show (see for instance example 1.4.1 in [1]) that the projection P_2 onto C_2 can be computed as follows.

$$(P_2\alpha)(e^{i\theta}) = \begin{cases} \dfrac{\alpha(e^{i\theta}) - g(e^{i\theta})}{|\alpha(e^{i\theta}) - g(e^{i\theta})|}\gamma_\infty + g(e^{i\theta}) & \text{if} \quad |\alpha(e^{i\theta}) - g(e^{i\theta})| > \gamma_\infty \\ \alpha(e^{i\theta}) & \text{otherwise} \end{cases} \tag{2.21}$$

where α is in L^2.

The projection P_3 onto C_3 is the nonlinear operator given by

$$P_3\alpha = \begin{cases} \dfrac{\alpha - f}{\|\alpha - f\|_2}\gamma_2 + f & \text{if} \quad \|f - \alpha\|_2 > \gamma_2 \\ \alpha & \text{otherwise} \end{cases} \tag{2.22}$$

where α is in L^2.

The projection P_{4i} onto the constraint set C_{4i} is computed by

$$P_{4i}\{\alpha_k\} = \begin{cases} \{\alpha_k\} - \dfrac{T_i\{\alpha_k\}^T - r_i - \gamma_t}{\|T_i\|^2}T_i & \text{if} \quad T_i\{\alpha_k\}^T > r_i - \gamma_t \\ \{\alpha_k\} - \dfrac{T_i\{\alpha_k\}^T - r_i + \gamma_t}{\|T_i\|^2}T_i & \text{if} \quad T_i\{\alpha_k\}^T < \gamma_t - r_i \\ \{\alpha_k\} & \text{otherwise} \end{cases} \tag{2.23}$$

where $\{\alpha_k\}$ is the sequence of the Fourier coefficients of α, and T_i is the ith row of the matrix

$$T = \begin{bmatrix} t_0 & 0 & \cdots & 0 & 0 & \cdots \\ t_1 & t_0 & \cdots & 0 & 0 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \\ t_{N-1} & t_{N-2} & \cdots & t_0 & 0 & \cdots \end{bmatrix} \tag{2.24}$$

In (2.24), $\{r_k\}$ and $\{t_k\}$ are the sequences of Fourier coefficients of the given functions r and t, respectively.

3. The Alternating Convex Projections Method

In this section we will review the alternating convex projection techniques in [5], [6], [15], [17]. Then, we will use these alternating projection techniques along with fast Fourier transform techniques to compute a solution to the $H_2/H_\infty/l_\infty$ feasibility problem in (2.16). To begin, let $\{C_1, C_2, ..., C_m\}$ be a family of closed convex sets in a Hilbert space $\mathcal{H}$ with inner product $(\cdot,\cdot)$, and C be the closed convex set defined by $C = C_1 \cap C_2 \cap \cdots \cap C_m$. Let P_i be the projection onto the set C_i and P_C be the projection operator onto C. Now define the composition operator Φ on $\mathcal{H}$ by $\Phi \triangleq P_n P_{n-1} ... P_1$. It follows that Φ is a contraction and the convex set C is precisely the set of fixed points of Φ, see [7]. The following result can be used to solve our feasibility problem.

Theorem 3.1 ([6]). *Let* x_o *be any vector in* $\mathcal{H}$. *If* C *is not empty, then* $\Phi^i x_o$ *converges weakly to a vector* x *in* C *as* $i \to \infty$. *If* C *is empty, the sequence* $\{x_i\}_{i=0}^{\infty}$ *of alternating projections* $x_1 = P_1 x_0$, $x_2 = P_2 x_1$, ..., $x_n = P_n x_{n-1}$, $x_{n+1} = P_1 x_n$, ... *does not converge.*

Recall that weak convergence guarantees that (x_i, z) converges to (x, z) for any z in $\mathcal{H}$, hence each Fourier coefficient of x converges to a limit. Under some mild assumptions on the constraint sets C_i, strong convergence (convergence in the norm) is guaranteed in Theorem 3.1, see [6], [15]. In our problems, we are interested to approximate x by a large but finite number of Fourier coefficients. Therefore, we can assume that $\{x_i\}_{i=0}^{\infty}$ lie ultimately in a finite-dimensional subspace $\mathcal{H}_N$ of $\mathcal{H}$. Hence strong convergence of $\{x_i\}_{i=0}^{\infty}$ to a point in $C \cap \mathcal{H}_N$ is guaranteed.

Notice that if C is empty, then $\Phi^i x_0$ will converge to a vector in C_n. However, the sequence $\{x_i\}_{i=0}^{\infty}$ will not converge. For example, if one has two disjoint balls in the plane, then the sequence $\{x_i\}_{i=0}^{\infty}$ will oscillate between two limit points, but $\Phi^i x_0$ will converge.

This algorithm can be implemented very easily in a computer, but may suffer from slow convergence (for example, consider the case where C_1 and C_2 are hyperplanes with "small" angles between them). One way to overcome this problem is to make use of some "directional information" in the algorithm. For simplicity of presentation we will state this algorithm for two convex sets, that is $C = C_1 \cap C_2$. This result is used to give us a fast algorithm to solve our feasibility problem [15].

Theorem 3.2 ([15]). *Assume that* $C = C_1 \cap C_2$ *is not empty. Let* x_o *be any vector in* $\mathcal{H}$ *and define the sequence* $\{x_i\}_{i=0}^{\infty}$ *by*

$$\left.\begin{aligned} & x_1 = P_1 x_o\,,\ \ x_2 = P_2 x_1\,,\ \ x_3 = P_1 x_2 \\ & x_4 = x_1 + \lambda_1 (x_3 - x_1)\,,\ \ \lambda_1 = \frac{\| x_1 - x_2 \|^2}{(\, x_1 - x_3,\, x_1 - x_2\,)} \end{aligned}\right\} \text{1}^{\text{st}} \text{ cycle}$$

$$\left.\begin{aligned} & x_5 = P_1 x_4\,,\ \ x_6 = P_2 x_5\,,\ \ x_7 = P_1 x_6 \\ & x_8 = x_5 + \lambda_2 (x_7 - x_5)\,,\ \ \lambda_2 = \frac{\| x_5 - x_6 \|^2}{(\, x_5 - x_7,\, x_5 - x_6\,)} \end{aligned}\right\} \text{2}^{\text{nd}} \text{ cycle} \tag{3.1}$$

$$\vdots$$

Then the sequence $\{x_i\}_{i=0}^{\infty}$ *converges to a vector* x *in* C. *The set* C *is empty if and only if the sequence* $\{x_i\}_{i=0}^{\infty}$ *does not converge.*

Notice that each cycle of the previous algorithm utilizes the projections onto the sets C_1 and C_2 to obtain a point which is closer to the intersection $C_1 \cap C_2$ than the point obtained from a cycle of the standard alternating projection algorithm of Theorem 3.1.

Remark 3.3: The result of Theorem 3.2 can be carried over to the case of three sets C_1, C_2 and C_3 as follows. Given a point x_0 in $\mathcal{H}$, compute the first cycle of the directional algorithm in Theorem 3.2 for two sets, say C_1 and C_2, to obtain a point x_1. Then, starting from x_1, compute the directional projection cycle for the sets C_2 and C_3 to obtain a point x_2. Follow the same procedure for the sets C_3 and C_1 and repeat the directional projection cycle for the pairs C_1, C_2, then C_2, C_3, and finally C_3, C_1, alternatively, until the sequence $\{x_i\}_{i=0}^{\infty}$ converges. This procedure can be extended to the general case of m sets in a similar fashion.

4. The Mixed $H_2/H_\infty/l_1$ Alternating Projections Algorithm

Numerically, for any rational function α in L^2, the projections $P_1\alpha$, $P_2\alpha$, $P_3\alpha$, and P_{4i} defined in section 2, can be computed approximately to any accuracy using fast Fourier transform (FFT) techniques. FFTs provide a very efficient means of transforming the function representation from time-domain to frequency-domain, and vice versa, to compute these projections. To this end, consider a rational function $\alpha = n/d$, where $n(z) = n_0 + n_1 z + \dots + n_{m-1} z^{m-1}$ and $d(z) = d_0 + d_1 z + \dots + d_{m-1} z^{m-1}$ are complex polynomials of $z = e^{i\theta}$. Choose, a number $K = 2^j$ (where j is integer and $K >> 2m$) of evaluation points around the unit circle, and compute the following vectors, using FFT s:

$$[\hat{n}_0, \hat{n}_1, \dots, \hat{n}_{K-1}] = \mathrm{FFT}([n_0, n_1, \dots, n_{m-1}], K) \tag{4.1}$$

$$[\hat{d}_0, \hat{d}_1, \dots, \hat{d}_{K-1}] = \mathrm{FFT}([d_0, d_1, \dots, d_{m-1}], K) \tag{4.2}$$

where $\hat{n}_k = n(e^{i2\pi k/K})$, $\hat{d}_k = d(e^{i2\pi k/K})$, $k = 0, 1, \dots, K-1$ and $\hat{x} = \mathrm{FFT}(x, K)$ denotes the fast Fourier transform of length K of a vector x padded with the appropriate number of zeros. The frequency-domain representation of the function is obtained by dividing n/d pointwise around the unit circle, to obtain

$$\alpha(e^{i2\pi k/K}) = \hat{\alpha}_k = \hat{n}_k / \hat{d}_k, \qquad (k = 0, 1, \dots, K-1)\,. \tag{4.3}$$

The Fourier coefficients of $\alpha = n/d$ can be approximated by computing the inverse FFT

$$[\alpha_0, \alpha_1, \dots, \alpha_{K-1}] = \mathrm{IFFT}([\hat{\alpha}_0, \hat{\alpha}_1, \dots, \hat{\alpha}_{K-1}], K) \tag{4.4}$$

where IFFT(y,K) denotes the inverse fast Fourier transform of length K of the vector y. Hence, we have

$$\alpha \approx \alpha_0 + \alpha_1 z + \alpha_2 z^2 + \dots + \frac{\alpha_{K-1}}{z} + \frac{\alpha_{K-2}}{z^2} + \dots \tag{4.5}$$

where the summation converges to α in L^2 norm, when $K \to \infty$. Notice that the first half of the coefficients $\alpha_0, \dots, \alpha_{K/2}$ correspond to the H_2 part of α while the last half of the coefficients $\alpha_{K-1}, \alpha_{K-2}, \dots, \alpha_{K/2+1}$ correspond to the $K_2 = L_2 \ominus H_2$ part of α.

Computation of the projection $P_1\alpha$: The Fourier coefficients of the projection $\beta = P_1\alpha$ are obtained by setting to zero the last $K/2 - 1$ elements of the vector $[\alpha_0, \alpha_1, \dots, \alpha_{K-1}]$ in (4.4), that is

$$\beta \triangleq P_1\alpha \approx \sum_{k=0}^{K/2} \alpha_k z^k \qquad (z = e^{i\theta}) \tag{4.6}$$

where the summation in (4.6) approaches $\beta = P_1\alpha$ in the L^2 norm when $K \to \infty$.

Computation of the projection $P_2\alpha$: The projection $\beta = P_2\alpha$ can be computed approximately in the frequency domain using Proposition 2.1, as follows:

$$\hat{\beta}_k = \begin{cases} \dfrac{\hat{\alpha}_k - \hat{g}_k}{|\hat{\alpha}_k - \hat{g}_k|}\gamma_\infty + \hat{g}_k & \text{if} \quad |\hat{\alpha}_k - \hat{g}_k| > \gamma_\infty \\ \hat{\alpha}_k & \text{otherwise} \end{cases} \tag{4.7}$$

for $k = 0, 1, \ldots, K-1$ where $\hat{\beta}_k$, $\hat{\alpha}_k$ and $\hat{g}_k$ are the pointwise evaluations of β, α and g, respectively, on the unit circle.

Computation of the projection $P_3\alpha$: The Fourier coefficients β_k of the projection $\beta = P_3\alpha$ can be computed using equation (2.16), as follows

$$\beta_k = \begin{cases} \dfrac{\alpha_k - f_k}{\|\alpha - f\|_2}\gamma_2 + f_k & \text{if} \quad \|f - \alpha\|_2 > \gamma_2 \\ \alpha_k & \text{otherwise} \end{cases} \tag{4.8}$$

for $k = 0, \ldots, K-1$, where α_k and f_k are the Fourier coefficients of the functions α and f, respectively. Using Parseval's formula, the L_2 distance $\|\alpha - f\|_2$ can be computed approximately as follows

$$\|f - \alpha\|_2^2 \approx \sum_{k=0}^{K-1} |\alpha_k - f_k|^2 . \tag{4.9}$$

Computation of the projection $P_{4i}\alpha$: The Fourier coefficients β_k of the projection $\beta = P_{4i}\alpha$ are computed as follows:

$$\beta_k = \begin{cases} \alpha_k - \dfrac{T_i\{\alpha_k\}^T - r_i - \gamma_t}{\|T_i\|^2} T_{ik} & \text{if} \quad T_i\{\alpha_k\}^T > r_i - \gamma_t \\ \alpha_k - \dfrac{T_i\{\alpha_k\}^T - r_i + \gamma_t}{\|T_i\|^2} T_{ik} & \text{if} \quad T_i\{\alpha_k\}^T < \gamma_t - r_i \\ \alpha_k & \text{otherwise} \end{cases} \tag{4.10}$$

where

$$\{\alpha_k\} = [\alpha_0, \alpha_1, \ldots, \alpha_{K-1}]$$

and T is the $N \times K$ matrix

$$T = \begin{bmatrix} t_0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ t_1 & t_0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ t_{N-1} & t_{N-2} & \cdots & t_0 & 0 & \cdots & 0 \end{bmatrix} \tag{4.11}$$

The solution to the $H_2/H_\infty/l_\infty$ Nehari feasibility problem (2.16) can be computed by applying the alternating projection algorithm in Theorem 3.1 or the directional projection algorithm in Theorem 3.2 to our FFT approximations to the projections P_1, P_2, P_3 and P_{4i}, $i = 0, 1, 2, \ldots, N$. Starting from any initial point x_0, the above algorithms are guaranteed to converge to a feasible solution q, if and only if the intersection $C_1 \cap C_2 \cap C_3 \cap C_{41} \cap \ldots \cap C_{4N}$ is nonempty. Now let $[q_0, q_1, q_2, \ldots] = \mathrm{IFFT}(\hat{q}_0, \hat{q}_1, \hat{q}_2, \ldots)$ be a solution to our $H_2/H_\infty/l_\infty$ feasibility problem obtained by the alternating projection technique. Then apply the Kalman-Ho algorithm [8] to the Markov parameters $\{q_i\}$ to obtain a minimal state-space realization $\{A, B, C\}$ for q, that is,

$q_i = CA^iB$ for $i \geq 0$. This realization provides a rational approximation for our solution q of the $H_2/H_\infty/l_1$ feasibility problem.

Remark 4.1. It is noted that the proposed techniques solve a finite-dimensional approximation of the exact infinite-dimensional mixed $H_2/H_\infty/l_\infty$ problem since a finite number of Fourier coefficients is considered. The proposed algorithms are guaranteed to converge to the intersection of the constraint sets C_1, C_2, C_3, C_{4i} and the finite-dimensional subspace spanned by these Fourier coefficients.

Remark 4.2. Given the rational functions $f_1, f_2, \ldots, f_{N_2}$, $g_1, g_2, \ldots, g_{N_\infty}$, $r_1, r_2, \ldots, r_{N_t}$ and $t_1, t_2, \ldots, t_{N_t}$ in L^∞, and the scalars $\gamma_{21}, \ldots, \gamma_{2N_2}$, $\gamma_{\infty 1}, \ldots, \gamma_{\infty N_\infty}$, and $\gamma_{t1}, \ldots, \gamma_{tN_t}$, our technique can be used to find a q in H_∞ (if it exists) that satisfies

$$\|f_j - q\|_2 \leq \gamma_{2j}, \quad \|g_k - q\|_\infty \leq \gamma_{\infty k} \quad \text{and} \quad \|r_k - tq\|_{l1} \leq \gamma_{tl} \tag{4.10}$$

for all $j = 1, 2, \ldots, N_2$, $k = 1, 2, \ldots, N_\infty$ and $l = 1, 2, \ldots, N_t$. The proposed techniques can be extended to solve the matrix mixed $H_2/H_\infty/l_1$ problem.

5. Numerical Example

Consider the mixed H_2/H_∞ feasibility problem

$$\|f - q\|_2 \leq \gamma_2 = 3$$

$$\|g - q\|_\infty \leq \gamma_\infty = 1.2$$

where f and g are defined by

$$f(z) = \frac{2}{z + 0.5}, \quad g(z) = \frac{z(z+4)}{(z-2)(z+0.5)}$$

Notice that f and g are not in H_∞. A solution to this problem was obtained using the directional alternating projections algorithm (3.1) of Theorem 3.2 and the FFT techniques of section 4. The algorithm converged to a feasible solution $h \in C_1 \cap C_2 \cap C_3$ in 12 cycles, for an error (sum of the distances from the constraint sets) less that 10^{-6} and a number of $K = 2^{10}$ FFT evaluation points. The required number of floating point operations (flops) to obtain a solution was approximately 2.1×10^6 and the required CPU time to execute the algorithm in MATLAB was 0.83 s in a Pentium 90 MHz processor. Figure 1 shows the magnitude plots of the functions f and g, and the feasible solution h provided by the alternating projection algorithm. Notice that $\|f - h\|_2 = 3$ and $\|g - h\|_\infty = 1.2$, that is the feasible solution h lies on the boundaries of the constraint sets C_2 and C_3. Figure 2 shows how the error decreases with the iteration number. For comparison, notice that the standard alternating projections algorithm of Theorem 3.1, converged to a feasible solution in 64 cycles, which required approximately 9.4×10^6 flops or 3.79 s of CPU time. The same problem was also solved using a deep-cut ellipsoid algorithm (e.g., see [5]). This algorithm converged to a feasible solution in 87 iterations, that required approximately 2.4×10^8 flops or 89.2 s of CPU time. The significantly smaller time required by the alternating projection methods to find a feasible solution is attributed to the use of FFTs and projections that exploit the geometry of the feasibility problem.

6. Conclusion

In this work we have used alternating projection algorithms to solve mixed $H_2/H_\infty/l_\infty$ problems. These algorithms are very simple to implement and utilize the simple geometry of the constraint sets, and FFTs to compute the required projections.

REFERENCES

[1] A.V. Balakrishnan, *Applied Functional Analysis*, Second Edition, Springer-Verlag, New York, 1981.

[2] D.S. Bernstein and W.M. Haddad, "LQG Control with an H_∞ Performance Bound: A Riccati Equation Approach," *IEEE Trans. Autom. Contr.*, Vol. 34, No. 3, pp. 293-305, 1989.

[3] S.P. Boyd, V. Balakrishnan, C.H. Barratt, N.M. Khraishi, X. Li, D.G. Meyer, and S.A. Norman, "A New CAD Method and Associated Architectures for Linear Controllers," *IEEE Trans. Automatic Control,*, Vol. 33, No. 3, 1988.

[4] S.P. Boyd and C.H. Barratt, *Linear Controller Design: Limits of Performance*, Prentice Hall, New Jersey, 1991.

[5] J.P. Boyle and R.L. Dykstra, "A Method for Finding Projections onto the Intersection of Convex Sets in Hilbert Space," Advances in Order Restricted Statistical Inference, Vol. 37, *Lecture Notes in Statistics*, Springer-Verlag, pp. 28-47, 1986.

[6] W. Cheney and A.A. Goldstein, "Proximity maps for convex sets," *Proc. Amer. Math. Soc.*, Vol. 12, pp. 448-450, 1959.

[7] M.A. Dahleh and B. Pearson," Minimization of a Regulated Response to a Fixed Input," *IEEE Trans. Automat. Control*, Vol. 33, pp. 924-930, 1988.

[8] A.A.H Damen, P.M.J Vanden Hof and A.K. Hajdasinski, "Approximate realization based upon an alternative to the Hankel matrix: the page matrix", *Systems Control Letters,* Vol. 2, pp. 202-208, 1982.

[9] C. Foias and A.E. Frazho, "Commutant Lifting and Simultaneous H^∞ and L^2 Suboptimization," *SIAM J. Math. Anal.,* Vol. 23, No. 4, pp. 984-994, 1992.

[10] C. Foias, A.E. Frazho and W.S. Li, "The Exact H^2 Estimate for the Central H^∞ Interpolant," *Integral Equations and Operator Theory,* to appear.

[11] A.E. Frazho, K.M. Grigoriadis and S. Kherat, "Alternating Projection methods for Mixed H^2 and H^∞ problems," *Proc. 31st Annual Allerton Conf. Commun., Control, Computing*, Urbana, Illinois, 1993.

[12] K. Glover, "All Optimal Hankel-Norm Approximations of Linear Multivariable Systems and their L^∞-Error bounds," *Int. J. Control,* Vol.39, No.6, pp. 1115-1193, 1984.

[13] K.M. Grigoriadis and R.E. Skelton, "Alternating Convex Projection Methods for Covariance Control Design," *Int. J. Control,* Vol. 60, No. 6, pp. 1083-1106, 1994.

[14] K.M. Grigoriadis and R.E. Skelton, "Low-order Control Design for LMI Problems Using Alternating Projections," *Automatica*, to appear.

[15] L.G. Gubin, B.T. Polyak and E.V. Raik, "The Method of Projections for Finding the Common Point of Convex Sets," *USSR Computational Mathematics and Mathematical Physics,* Vol. 7, pp. 1-24, 1967.

[16] W.M. Haddad, D.S. Bernstein and D. Mustafa, "Mixed H_2/H_∞ Regulation and Estimation: The Discrete-Time Case," *Systems Control Lett.*, Vol.16, pp. 235-247, 1991.

[17] S-P. Han, "A Successive Projection Method", *Mathematical Programming,* Vol. 40, pp. 1-14, 1988.

[18] I. Kaminer, P.P. Khargonekar and M.A. Rotea, "Mixed H_2/H_∞ Control for Discrete-Time Systems via Convex Optimization," *Automatica,* Vol. 29, No. 1, pp.57-70, 1993.

[19] P.P. Khargonekar and M.A. Rotea, Mixed H^2/H^∞ Control: A Convex Optimization Approach, *IEEE trans. Autom. Control,* Vol. 36, No 7, pp. 824-837, 1991.

[20] A. Sideris, "Single-input-Single-output H_∞-control with Time Domain Constraints," *Automatica*, Vol. 29m No. 4, pp. 969-983, 1993.

[21] M. Sznaier, "An Exact Solution to General SISO Mixed H_2/H_∞ Problems via Convex Optimization," *Proc. Amer. Control Conf*, San Francisco, California, June 1993.

[22] M. Sznaier, "A Mixed L_∞/H_∞ Optimization Approach to Robust Controller Design," *SIAM J. Control and Optim.*, Vol. 33, No. 4, pp. 1086-1101, 1995.

[23] J. Von Neumann, *Functional Operators,* Vol. II, Ann. Math. Studies No.22, Princeton, New Jersey, 1950.

[24] D.C. Youla and H. Webb, "Image restoration by the method of convex projections: Part 1 - Theory," *IEEE Trans. Medical Imag.,* Vol. 1, No. 2, pp. 81-94, 1982.

[25] N. Young, *An Introduction to Hilbert Space,* Cambridge University Press, Cambridge, NY, 1988.

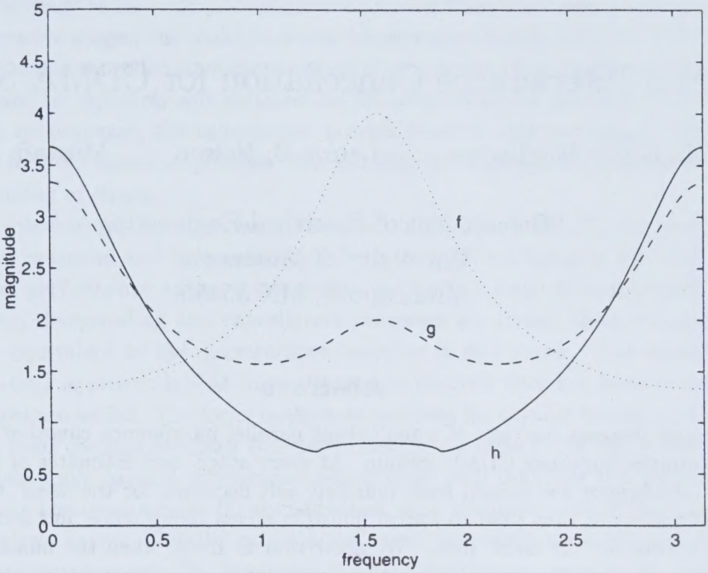


Fig. 1: Magnitude plots

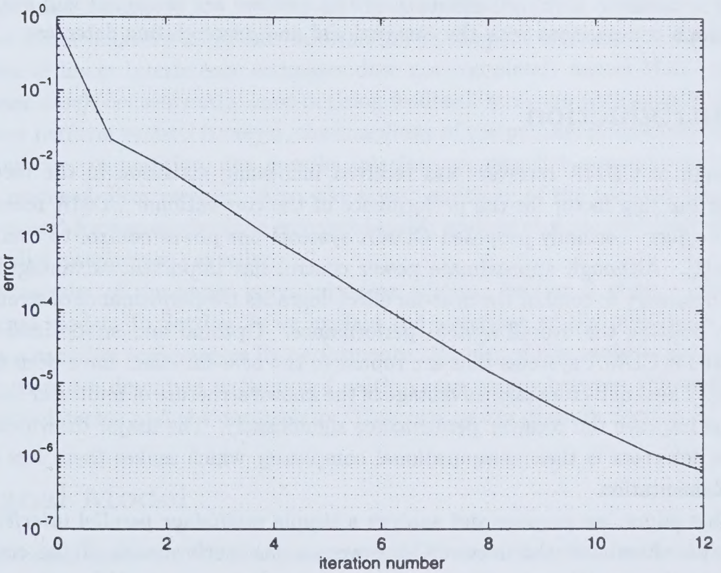


Fig. 2: Error vs. iteration number

Parallel Interference Cancellation for CDMA Systems

Vafa Ghazi-Moghadam Laurie B. Nelson Mostafa Kaveh

Department of Electrical Engineering
University of Minnesota
Minneapolis, MN 55455

Abstract

This paper presents analysis of a multistage parallel interference canceller receiver for a synchronous multiuser CDMA system. At every stage, new estimates of the multiple access interference are formed from infinitely soft decisions for the users' data. These new estimates are then used to cancel multiple access interference and form new decision statistics for the users' data. We show that in limit, when the number of stages increases infinitely, this receiver has the performance of a decorrelating detector. For an arbitrary finite number of stages, closed form expressions for the probability of error and the asymptotic multiuser efficiency of this receiver are presented and evaluated for performance comparisons with the conventional and decorrelating detectors.

1 Introduction

The design of CDMA receivers has received increasing attention in the recent years. A major limiting factor for the performance of the conventional CDMA receiver is the *near-far effect.* Formally proposed CDMA systems use power control to overcome this problem[1]. Although approximate power control has important advantages, the fine control necessary to combat the near-far effect degrades the performance of stronger users, thereby limiting the overall system performance. Optimal and suboptimal *multiuser* detectors for CDMA systems that are robust to the near-far effect have been considered in [2]-[5]. These detectors take advantage of the known structure of multiuser interference, and thus improve the receiver performance significantly. The major drawback of many of these detectors is their computational complexity which makes them less attractive for implementation.

In this paper, we propose and analyze a simple *multistage parallel interference canceller.* We show that this receiver improves on the performance of the conventional detector without adding significant complexity to that receiver. We assume a bit synchronous system to simplify the analysis; conceptually, the receiver and its analysis can be easily extended to an asynchronous system.

This work was supported in part by the National Science Foundation under Grant MIP-9202081.

The initial stage of the receiver consists of a bank of filters matched to each user's signal. At successive stages, the multiple-access interference in each matched-filter output is re-estimated and cancelled from the matched-filter outputs. The interference estimates are determined by *infinitely soft decisions* on the outputs of the previous stage cancellation. As a consequence, the cancellation process requires only the signal correlations and not the received signal amplitudes. The parallel cancellation can be repeated for an arbitrary number of stages.

Parallel interference cancellers were discussed in [6]-[8]. In [6]-[7], the interference estimates are re-spread and then subtracted from the received signal in the continuous-time domain; performance analyses are carried out for one stage of cancellation. Since the spreading, de-spreading and cancellation processes are linear, these structures are functionally equivalent to the discrete-time canceller in this paper. One advantage of the discrete-time approach is that the performance analysis does not become unwieldy as more stages are added. The focus in [8] is on analysis for parallel interference cancellation using hard decision feedback (similar to multistage detector in [4]); some limited results for one stage parallel interference cancellation with infinitely soft decisions are also included. The results in [6]-[8] are consistent with the more general (and arguably simpler) analysis herein. Parallel cancellation using hard-decisions is also used in the *multistage multiuser receiver* [4]. Replacing the hard-decision elements in the multistage structure of [4] with linear *infinitely soft* decisions leads to the parallel interference canceller considered herein. A key motivation for using infinitely soft decisions is that the receiver no longer requires amplitude information to compute estimates of the MAI.

The use of linear interference estimates does not completely cancel MAI (except in the two-user case); consequently, hard-decision feedback receivers with amplitude information may perform better. However, the simplicity of the parallel interference canceller is advantageous; in addition, the parallel interference canceller outperforms the conventional matched-filter receiver. Also, the linear structure of the receiver allows for a thorough closed-form analysis of the error probability and asymptotic multiuser efficiency of the parallel interference canceller.

The remainder of the paper is organized as follows. Section 2 gives a description of the signal model. In Section 3, we explain the operation of the multistage parallel interference canceller and analyze its performance. Results of the analysis for a detector with four users are described in Section 4, with a comparison between this system, the conventional detector and the decorrelator. Conclusions are given in Section 5.

2 Signal Model

In a synchronous multiuser CDMA system, where K users transmit simultaneously over a passband channel, the received signal can be modeled as:

$$r(t) = \sum_{k=1}^{K} A_k b_k s_k(t) + \sigma n(t) \tag{1}$$

where A_k, b_k and $s_k(t)$ are the amplitude, data bit and the normalized signature waveform of duration T for the k^{th} user, respectively. The term $n(t)$ is the unit spectral density additive Gaussian noise. The channel bandwidth is assumed to be large enough so that intersymbol interference can be ignored. Sending the received signal through a bank of filters matched to the signature waveforms of the K users, and sampling at time T, results in the following discrete-time model:

$$\mathbf{y} = \mathbf{RAb} + \mathbf{n} \tag{2}$$

where $\mathbf{y} = [y_1 \cdots y_K]^T$ is the vector of the outputs of the matched filters, $\mathbf{b} = [b_1 \cdots b_K]^T$ is the set of transmitted bits and $\mathbf{n} = [n_1 \cdots n_K]^T$ is a Gaussian noise vector with covariance matrix $\sigma^2\mathbf{R}$. The matrix $\mathbf{A}$ is diagonal with the amplitudes of the K users as its elements and $\mathbf{R} = [\rho_{ij}]$ is the positive definite matrix of crosscorrelations between the signature waveforms:

$$\rho_{ij} = \int_0^T s_i(t)s_j(t)dt \qquad\qquad i,j = 1, \cdots, K.$$

3 Multistage canceller

The conventional detector for a CDMA system processes each of the outputs of the matched filter independently, yielding the following set of decisions:

$$\hat{\mathbf{b}}^c = sgn(\mathbf{y}) \tag{3}$$

This detector treats the MAI as Gaussian noise and does not take advantage of the structure of multiple access signal constellation. It is well known that this detector is not *near-far resistant*; i.e., its performance degrades significantly when the interfering users dominate the desired user in energy. In this section we develop a simple multistage detector that provides better performance than the conventional detector, in most practical cases. At each stage, the receiver uses infinitely soft decisions on the outputs of the previous stage to estimate and cancel the multiple access interference from the matched filter outputs . At the first stage, the following set of new estimates of the K users' transmitted signals are formed, using the outputs of the matched filters:

$$y_k^{(1)} = y_k - \sum_{\substack{i=1 \\ i \neq k}}^{K} \rho_{ki} y_i, \qquad\qquad k = 1, \cdots, K. \tag{4}$$

No decisions (i.e., infinitely soft decisions) are made on the outputs of the matched filter before performing the cancellation. Representing the new outputs as a vector, we can write

$$\mathbf{y}^{(1)} = \mathbf{y} - (\mathbf{R} - \mathbf{I})\mathbf{y} = (2\mathbf{I} - \mathbf{R})\mathbf{y} = \mathbf{L}^{(1)}\mathbf{y}. \tag{5}$$

As it can be seen, this parallel cancellation results in a linear detector. At the second stage, the cancellation is performed by using the new set of estimates, $\mathbf{y}^{(1)}$:

$$\mathbf{y}^{(2)} = \mathbf{y} - (\mathbf{R} - \mathbf{I})\mathbf{y}^{(1)} = (3\mathbf{I} - 3\mathbf{R} + \mathbf{R}^2)\mathbf{y} = \mathbf{L}^{(2)}\mathbf{y}. \tag{6}$$

In general, for the n^{th} stage, we can write

$$\mathbf{y}^{(n)} = \mathbf{L}^{(n)}\mathbf{y}. \tag{7}$$

After some algebraic manipulation, $\mathbf{L}^{(n)}$ is found to be

$$\mathbf{L}^{(n)} = \mathbf{R}^{-1}[\mathbf{I} - (\mathbf{I} - \mathbf{R})^{n+1}]. \tag{8}$$

The equation above shows that the multistage parallel canceller can be considered as a combination of a decorrelator and a residual term. Now, we show that under certain conditions, the residual term in the brackets diminishes as the number of stages increase; i.e.,

$$\lim_{n\to\infty} (\mathbf{I} - \mathbf{R})^{n+1} = \mathbf{0} \tag{9}$$

This means that the multistage parallel canceller will perform (in the limit) like the decorrelator proposed in [3]. In general, for a matrix $\mathbf{C}$, the following holds[9]:

$$\lim_{n\to\infty} \mathbf{C}^n = \mathbf{0} \Longleftrightarrow r(\mathbf{C}) < 1, \tag{10}$$

where $r(\mathbf{C})$ is the *spectral radius* of $\mathbf{C}$; $r(\mathbf{C}) = \max\{|\lambda| : \lambda \text{ is an eigenvalue of } \mathbf{C}\}$, (Thus, $r(\mathbf{I} - \mathbf{R}) < 1$ is the necessary and sufficient condition for the Eq. 9). In general, there is no closed form solution for the spectral radius of a matrix. However, for any matrix norm $\|\cdot\|$, $r(\mathbf{C}) \leq \|\mathbf{C}\|$; in addition, for a symmetric matrix, $r(\mathbf{C}) = \|\mathbf{C}\|_2$ where $\|\mathbf{C}\|_2$ is the 2-norm of matrix $\mathbf{C}$ [9].

For a $K \times K$ matrix, the following relationship holds[10]:

$$\|\mathbf{C}\|_2 \leq \|\mathbf{C}\|_\infty = \max_{1\leq i\leq K} \sum_{j=1}^{K} |c_{ij}| \tag{11}$$

Thus, we can use the infinite-norm, which can be calculated in closed form, as an upper bound for the spectral radius. For our example, $\mathbf{C} = \mathbf{I} - \mathbf{R}$ is a symmetric matrix with zero diagonal elements; i.e.,

$$c_{ij} = \begin{cases} 0 & i = j \\ -\rho_{ij} & \text{otherwise} \end{cases}. \tag{12}$$

So the sufficient (not necessary) condition to satisfy Eq. 9 is

$$\max_{1\leq i\leq K} \sum_{\substack{j=1 \\ j\neq i}}^{K} |\rho_{ij}| < 1. \tag{13}$$

This condition is independent of the amplitudes of the received signals from different users.

In multiuser detection, a measure which is useful in analyzing a detector is the *asymptotic multiuser efficiency*, which examines the performance of a detector in low noise situations and is defined as[2]:

$$\eta_k = sup\{0 \leq r \leq 1; \lim_{\sigma\to 0} P_k(\sigma)/Q(\frac{\sqrt{r}A_k}{\sigma}) < +\infty\} \tag{14}$$

For practical error rates ($\sigma \neq 0$), it can be seen that $P_k(\sigma) \approx Q(\sqrt{\eta_k}\text{SNR})$, which implies that when $\eta_k > 0$, arbitrarily small probabilities of error can be achieved by increasing the signal to noise ratio. The asymptotic multiuser efficiency of a conventional detector operating in a power control mode is $(A_1 = A_2 = \cdots = A_K)$

$$\eta_k = \max^2\{0, 1 - \sum_{\substack{i=1 \\ i \neq k}}^{K} |\rho_{ik}|\}. \tag{15}$$

For η_k to be greater than zero for all users, we must have

$$\sum_{\substack{i=1 \\ i \neq k}}^{K} |\rho_{ik}| < 1 \qquad \text{for all } k = 1, \cdots, N,$$

which is equivalent to the condition derived in Eq. 13. This means that if the users' spread spectrum signals are chosen such that, with power control a conventional detector can operate with a certain quality of service, then the proposed scheme will converge to a decorrelator as the number of stages increases. As stated above, the condition in Eq. 13 is derived from an upper bound for the spectral radius $r(\mathbf{I} - \mathbf{R})$. For the case where all the off-diagonal terms of $\mathbf{R}$ are the same (i.e., when the crosscorrelations between every two signature waveforms are equal) then the upper bound is tight. For this case, $r(\mathbf{I} - \mathbf{R}) = (K-1)\rho$, where K is the number of users, and ρ is the crosscorrelation of two distinct signature waveforms.

3.1 Performance analysis of the scheme

The k^{th} user error probability for a conventional detector (without power control) is found to be

$$\begin{aligned} P_k^c = P[y_k < 0 | b_k = 1] &= 2^{1-K} \sum_{\substack{\mathbf{b} \in \{-1,1\}^K \\ b_k = 1}} Q\left(\frac{A_k + \sum_{i \neq k} \rho_{ik} A_i b_i}{\sigma}\right) \\ &= 2^{1-K} \sum_{\substack{\mathbf{b} \in \{-1,1\}^K \\ b_k = 1}} Q\left(\frac{(\mathbf{RAb})_k}{\sigma\sqrt{(\mathbf{R})_{kk}}}\right). \end{aligned} \tag{16}$$

For the multistage parallel interference canceller, the error probability of the k^{th} user is similarly derived as

$$P_k^{(n)} = 2^{1-K} \sum_{\substack{\mathbf{b} \in \{-1,1\}^K \\ b_k = 1}} Q\left(\frac{(\mathbf{L}^{(n)}\mathbf{RAb})_k}{\sigma\sqrt{(\mathbf{R}^{(n)})_{kk}}}\right), \tag{17}$$

where $\sigma^2\mathbf{R}^{(n)} = \sigma^2\mathbf{L}^{(n)}\mathbf{R}\mathbf{L}^{(n)}$ is the covariance matrix of the Gaussian noise vector at the output of the n^{th} stage, and $\mathbf{L}^{(n)}$ is the symmetric matrix defined in Eq. 8. From Eqs. 16-17, asymptotic multiuser efficiencies for both detectors are given by

$$\begin{aligned} \eta_k^c &= \min_{\substack{\mathbf{b} \in \{-1,1\}^K \\ b_k = 1}} \max^2\{0, \frac{(\mathbf{RAb})_k}{\sqrt{(\mathbf{R})_{kk}}}\} \\ \eta_k^{(n)} &= \min_{\substack{\mathbf{b} \in \{-1,1\}^K \\ b_k = 1}} \max^2\{0, \frac{(\mathbf{L}^{(n)}\mathbf{RAb})_k}{\sqrt{(\mathbf{R}^{(n)})_{kk}}}\}. \end{aligned} \tag{18}$$

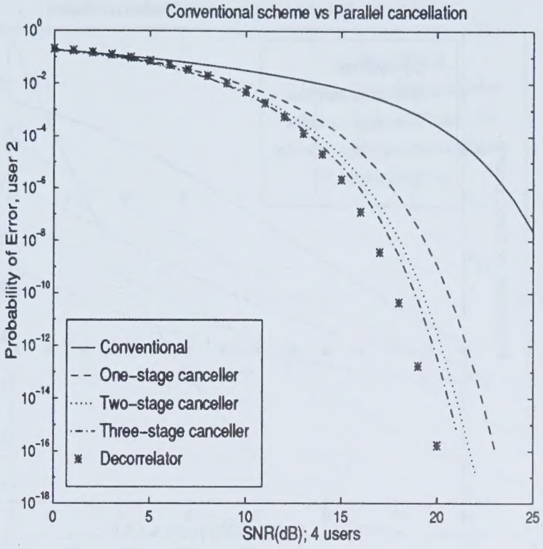


Figure 1: Probability of Error vs SNR of the desired user

4 Analysis Results

For an example with 2 users, the probabilities of error for the conventional detector and a one-stage canceller are given by

$$\begin{aligned} P_1^c &= \frac{1}{2}Q\left(\frac{A_1}{\sigma}(1-\rho\frac{A_2}{A_1})\right)+\frac{1}{2}Q\left(\frac{A_1}{\sigma}(1+\rho\frac{A_2}{A_1})\right) \\ P_1^{(1)} &= Q\left(\frac{A_1}{\sigma}\sqrt{1-\rho^2}\right), \end{aligned} \tag{19}$$

respectively. Note that for one-stage canceller, the probability of error is the same as that for the decorrelator.

To demonstrate the performance of the multistage canceller and compare it with the conventional detector and the decorrelating detector, we use as an example, a set of signature sequences based on Gold codes of length seven, which have also been used in [4] and [5]. We assume four active users, for which the crosscorrelation matrix is:

$$\mathbf{R}=\frac{1}{7}\begin{pmatrix} 7 & -1 & 3 & -1 \\ -1 & 7 & 3 & 3 \\ 3 & 3 & 7 & -1 \\ -1 & 3 & -1 & 7 \end{pmatrix}.$$

Fig. 1 compares the probability of error curves for conventional, multistage(1-3 stages) and decorrelating detectors, as a function of signal-to-noise ratio $\frac{A_k^2}{\sigma^2}$. All four

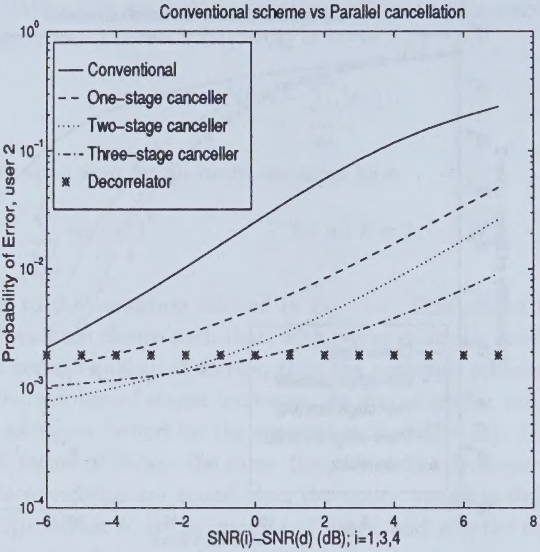


Figure 2: Probability of Error vs SNR of the interfering users, 4 users

users are assumed to have equal powers at the receiver. The multistage canceller performs much better than conventional detector for probability of errors of practical interest (e.g., for a one stage canceller and $P_1^{(1)} = 10^{-4}$, the SNR required for the desired user is 6 dB less than what is required for the conventional detector). In addition, it can be seen that the performance of the multistage canceller approaches that of the decorrelator as the number of stages increase.

Fig. 2 shows the probability of error curves for the same detectors, as a function of the interfering users' SNRs. The SNR for the desired user is 11 dB, and the SNRs for the interfering users vary uniformly from 5 to 18 dB. We can see that as the number of stages are increased, the multistage canceller becomes less vulnerable to near-far problem than a conventional detector, and approaches the decorrelator, which is a near-far resistant receiver. For weak interference, the parallel canceller performs better than the decorrelator.

In Fig. 3, the asymptotic multiuser efficiencies of the aforementioned detectors are shown as a function of the ratio of interference amplitude to the desired user's signal amplitude. The asymptotic multiuser efficiency of a multistage canceller degrades more slowly than that of the conventional detector, and it converges to the asymptotic multiuser efficiency of the decorrelator as the number of stages increase.

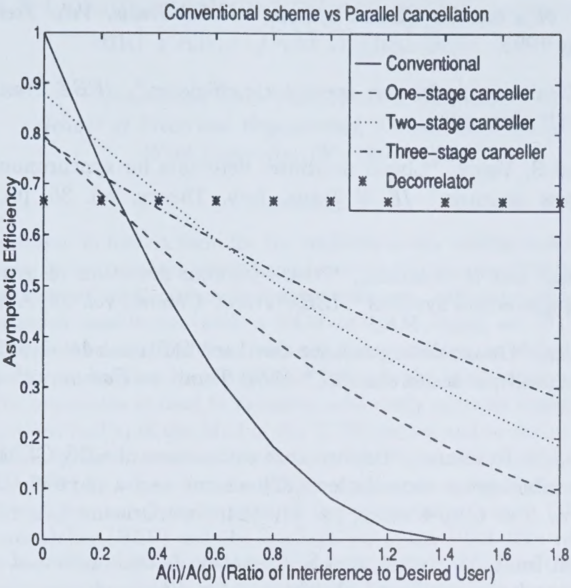


Figure 3: Asymptotic multiuser efficiency vs the ratio of interfering and desired users

5 Conclusion

This paper presents a simple and thorough analysis of an iterative parallel interference cancellation receiver for synchronous CDMA. Unlike nonlinear detectors using decision feedback (with hard or soft limiting), the parallel interference canceller does not require knowledge of the signal amplitudes. Simulations and analytical results showed that its performance is comparable to that of the decorrelator as the number of stages becomes large. In particular, a sufficient condition for convergence of the parallel interference canceller to the decorrelating detector was derived.

In the presence of moderate to strong interference, the parallel interference canceller provides improvement over the conventional receiver, although its performance (for a small number of stages) is somewhat worse than the decorrelator. For applications where the set of signature sequences (or active users) changes often, the parallel interference canceller may have some implementational advantages over the decorrelator, since its structure could be easily updated to reflect a change in the set of active users, whereas the decorrelator would require recomputing a matrix inversion in this situation. The parallel interference canceller also has implementation advantages for asyncrhonous channels, for which the decorrelator has infinite impulse response.

References

[1] K. Gilhousen, I. Jacobs, R. Padovani, A. Viterbi, L. Weaver and C. Wheatley "On

the capacity of a cellular CDMA system," *IEEE Trans. Veh. Tech.*, vol. 40, pp. 303-312, May 1991.

[2] S. Verdú, "Optimum multi-user asymptotic efficiency", *IEEE Trans. Comm.*, vol. 34, pp. 890-897, Sep. 1986.

[3] R. Lupas and S. Verdú, "Linear multiuser detectors for synchronous code-division multiple-access channels," *IEEE Trans. Info. Theory*, vol. 35, pp. 123-136, Jan. 1989.

[4] M.K. Varanasi and B. Aazhang, "Near-optimum detection in synchronous code-division multiple-access systems," *IEEE Trans. Comm.*, vol. 38, April 1990.

[5] A. Duel-Hallen, "Decorrelating decision-feedback multiuser detector for synchronous code-division multiple-access channel," *IEEE Trans. on Comm.*, vol. 41, pp. 285-290, Feb. 1993.

[6] P. Patel and J.M. Holtzman, "Performance comparison of a DS/CDMA system using a successive interference cancellation (IC) scheme and a parallel IC scheme under fading ," *Proc. Int. Conf Comm.*, pp. 510-514, New Orleans, LA, 1994.

[7] R. Kohno, H. Imai, M. Hatori, and S. Paspathy, "Combination of an adaptive array antenna and a canceller of interference for direct- sequence spread-spectrum multiple-access system," *IEEE Journal on Selected Areas in Communications*, vol. 8, pp. 675-682, May 1990.

[8] D. Divsalar and M. K. Simon, "Improved CDMA performance using parallel interference cancellation," *Proc. MILCOM*, pp. 911-917, 1994.

[9] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, New York, 1985.

[10] G.W. Stewart and J. Sun, *Matrix Perturbation Theory*, Academic Press, Boston, 1990.

TCM/SSMA Communication Systems with Cascaded-Sequences and PAM/QAM Signal Sets *

Tsao-Tsen Chen and James S. Lehnert
School of Electrical Engineering, Purdue University
West Lafayette, IN 47907-1285

Abstract

An expression in matrix form for the multiple-access interference (MAI) in an asynchronous direct-sequence spread-spectrum multiple-access (DS/SSMA) communication system with cascaded-sequence schemes, arbitrary chip waveforms, and trellis-coded modulation with a PAM or QAM signal set (TCM/PAM or TCM/QAM), is obtained. TCM provides significant coding gain while the cascaded-sequence schemes decrease the correlation between the MAI of adjacent data intervals. The expression is used to calculate arbitrarily accurate cumulative distribution functions (cdf's) of the MAI in the TCM system and to derive an accurate approximation of the variances of the MAI. It also helps us to illustrate some properties of the MAI by separating contributing parameters into different matrices. We also investigate an approximation of the upper union bound (UUB) on the bit error probability (BEP) and the applicability of the UUB. The results of this analysis reveal which code parameters have the greatest impact on the average error performance and provide analytical tools for use in preliminary system design. The effects of the length of the cascaded-sequences and the number of interfering transmitters are illustrated.

1 Introduction

A cascaded-sequence (CV) scheme (we assign to each user one or two sets of V spreading sequences) is of interest because it decreases the correlation between the MAI of adjacent channel symbols. In [1], the expression for the MAI is not studied for the general CV schemes and arbitrary chip waveforms. In [3] and [5], methods to obtain arbitrarily accurate cdf's of the MAI with arbitrary chip waveforms are given, but multi-level signal sets and coding are not studied. In this paper, we derive an expression in matrix form for the MAI of the receiver when the general CV schemes, arbitrary chip waveforms, and TCM/PAM or TCM/QAM are used. The columns of the matrix represent all the possible specifications of the MAI. Computational complexity can be greatly reduced by transforming the matrix into the standard forms in [7] before applying the techniques used in [3] and [5] to obtain the accurate pairwise error probability (PEP). The expression also separates the effects on the MAI from spreading sequences, chip waveforms, error weight sequences, and modulated sequences of interfering users into different matrices. This helps us to illustrate some useful properties of the MAI probability density functions (pdf's). Our result shows that the CV schemes can greatly reduce the PEP's until the length of the cascaded sequence becomes greater than that of the error path under consideration. This expression is used to derive an accurate approximation of the MAI variance, which shows that this variance is proportional to the squared Euclidean distance of the error sequence under consideration. The reduction in computational complexity is also applicable to calculations of the Gaussian approximation of the MAI. Previous work on the variance of the MAI [2], [4] is a special case of these general results. Through the

---

*This research was supported in part by the U.S. Army Research Office under grant DAAH04-95-I-0246.

expression for the approximation of the variance, we identify the key parameters that are useful in the design of the spreading sequences and chip waveforms for the CV schemes.

Performance of TCM systems over the additive white Gaussian noise (AWGN) channel can be evaluated using different methods. A widely-used technique is the Chernoff bound combined with the modified generating function [8] for many trellis codes which are based on set partitioning [10]. If the signal to thermal noise ratio is large, the PEP decays very quickly in squared Euclidean distances. That is, we need only consider the dominating PEP's from the first few error paths. The situation becomes more complicated when we assume that the channel noise is a combination of AWGN and the MAI. In [1], the computation of the BEP considers only the first error path. Our results show that the BEP's due to error paths other than the 1st one may not be negligible. We thus neglect the AWGN and focus on the MAI in order to get a clear picture of how the MAI affects the error performance of the TCM system. We will see that some key parameters play the same role as the signal to thermal noise ratio in the system with the AWGN only. For short error paths, the computation of the exact PEP's is moderately complex. However, the complexity increases rapidly with error path length L and number of spreading sequences V. In such a situation, it is impractical to calculate the exact PEP's for all the error paths. Our approach is to calculate the exact PEP's for the first v error paths and use a Gaussian approximation to get the PEP's for the remaining error paths. The exact v PEP's are used to obtain a lower bound (LB) on the UUB. An exponential bound (EB) is then used, together with the modified generating function and LB, to obtain an upper bound (UB) on the UUB. A Gaussian bound (GB) is also illustrated when we use a Gaussian approximation for all the MAI. Our results show that when the number of simultaneous users K is small or moderate, or equivalently, the variance of the MAI is small, then the error performance of the TCM system is better than the uncoded system. When K is large, the UB and GB will merge and become quite loose. Since the MAI is not exactly Gaussian distributed, the above approach is used only to illustrate how some code parameters affect the system performance.

2 System Model

The system model that we examine is based on [1]. There are K transmitter/receiver pairs sharing a common AWGN channel. In this section, we use the more general notations for the QAM system. If just the in-phase sections of the QAM system are examined, a description of a PAM system is obtained.

2.1 Transmitter

The model for the TCM/SSMA transmitter with a QAM signal set is shown in Fig. 1. If a rectangular pulse is defined by $p_T(t) = 1$ for $0 \leq t \leq T$ and $p_T(t) = 0$, otherwise, then the kth data signal can be expressed as $b^k(t) = \sum_{j=-\infty}^{\infty} b_j^{(k)} p_T(t - jT)$, where the sequence $\{b_j^{(k)}\}$ is the binary data sequence of the kth transmitter and is a sequence of independent and identically distributed random variables such that $\Pr\{b_j^{(k)} = -1\} = \Pr\{b_j^{(k)} = +1\} = 1/2$. The input data sequence is divided into blocks of input data with n data bits in each block. For n input bits from the data source, n_1 of these bits select a coset of the signal constellation through a rate $n_1/n_1 + 1$ convolutional encoder, while the remaining n_2 select a signal point within the given coset. We define the in-phase baseband coded signal $\tilde{b}_I^k(t)$ of the kth user by $\tilde{b}_I^k(t) = \sum_{l=-\infty}^{\infty} \tilde{b}_{I,l}^{(k)} p_T(t - lT)$, and the quadrature baseband coded signal $\tilde{b}_Q^k(t)$ of the kth user by $\tilde{b}_Q^k(t) = \sum_{l=-\infty}^{\infty} \tilde{b}_{Q,l}^{(k)} p_T(t - lT)$, where the sequence $\{\tilde{b}_{I,l}^{(k)}, \tilde{b}_{Q,l}^{(k)}\}$ is the output sequence from the trellis encoder of the kth transmitter. For an M^2-ary QAM signal set, we have $\{\tilde{b}_{I,l}^{(k)}, \tilde{b}_{Q,l}^{(k)}\} \in \{\pm(\mathrm{M}-1), \pm(\mathrm{M}-3), \ldots, \pm 1\} \times \{\pm(\mathrm{M}-1), \pm(\mathrm{M}-3), \ldots, \pm 1\}$ for each l. The average signal energy of an M^2-ary QAM signal set is $E_{Q,s} = 2 \times \frac{M^2-1}{3} T E_c P \triangleq 2\tilde{M} T E_c P$.

As for the spreading sequences, we assume that these direct sequences are deterministic. In the CV scheme, we assign to each transmitter/receiver pair two sets of V spreading sequences and use these sequences sequentially. For the kth user, we denote the spreading sequences used for in-phase and quadrature channels by $\{\alpha_{I,1}^{(k)}, \alpha_{I,2}^{(k)}, \ldots, \alpha_{I,V}^{(k)}\}$ and $\{\alpha_{Q,1}^{(k)}, \alpha_{Q,2}^{(k)}, \ldots, \alpha_{Q,V}^{(k)}\}$ respectively. Each sequence $\alpha_{B,j}^{(k)}$, where $j \in \{1, 2, \ldots, V\}$ and $B \in \{I, Q\}$, consists of N chip symbols; i.e., $\alpha_{B,j}^{(k)} = (\alpha_{B,j,0}^{(k)}, \alpha_{B,j,1}^{(k)}, \ldots, \alpha_{B,j,N-1}^{(k)})$, where $\alpha_{B,j,i}^{(k)} \in \{-1, 1\}$ for $i = 0, 1, \ldots, N-1$. If the chip waveform $\psi(t)$ is defined to be time-limited to the interval $[0, T_c)$, the spreading sequence of the kth transmitter can be expressed as $\alpha_{B,j}^{(k)}(t) = \sum_{i=0}^{N-1} \alpha_{B,j,i}^{(k)} \psi(t - iT_c)$. We assume that each element in the sequence $\{\tilde{b}_{I,l}^{(k)}, \tilde{b}_{Q,l}^{(k)}\}$ is encoded with N chips; i.e., $T = NT_c$. The normalized energy of the chip waveform $\psi(t)$ is given by $E_c = \frac{1}{T_c} \int_0^{T_c} \psi^2(t) dt$. If we define the in-phase and quadrature transmitted signals of the kth user during the lth signaling interval by $\sigma_I^k(\tilde{b}_{I,l}^{(k)}, \alpha_{I,<l>}^{(k)}, t) = \sqrt{2P} \cos(\omega_c t + \phi_k) \tilde{b}_{I,l}^{(k)} \alpha_{I,<l>}^{(k)}(t)$, and $\sigma_Q^k(\tilde{b}_{Q,l}^{(k)}, \alpha_{Q,<l>}^{(k)}, t) = \sqrt{2P} \sin(\omega_c t + \phi_k) \tilde{b}_{Q,l}^{(k)} \alpha_{Q,<l>}^{(k)}(t)$, where $<l> = l$ modulo V, P is the power of the transmitted signal and ϕ_k models the phase of the kth transmitter, then the transmitted signal of the kth user over all signaling intervals can be expressed as $S^k(t) = \sum_{l=-\infty}^{\infty} \sigma_I^k(\tilde{b}_{I,l}^{(k)}, \alpha_{I,<l>}^{(k)}, t - lT) + \sigma_Q^k(\tilde{b}_{Q,l}^{(k)}, \alpha_{Q,<l>}^{(k)}, t - lT)$.

2.2 Channel

We assume that the thermal noise $n(t)$ is an additive white Gaussian process with zero mean and two-sided power spectral density $N_o/2$. The received signal for all receivers is $r(t) = \sum_{k=1}^{K} S^k(t - \tau_k) + n(t)$, where τ_k is the time delay parameter that accounts for propagation delay and the lack of synchronism between the transmitters.

2.3 Receiver

Because of the symmetry of the model, we need consider only the receiver that is listening to the first transmitter. Also, since only relative delays and phases are important, we set $\tau_1 = \theta_1 = 0$. The properties of a SSMA system and the stationarity of the noise $n(t)$ permit us to restrict attention to time delays modulo VT and phase angles modulo 2π. Hence, for $k \neq 1$, we model the delay τ_k and the phase angle θ_k as random variables that are uniformly distributed on $[0, VT)$ and $[0, 2\pi)$ respectively.

We use correlation receivers (see Fig. 2) to demodulate the received signal. The receiver of the first transmitter forms two decision statistics at interval l, the in-phase decision statistic $Z_{I,l}$ and the quadrature decision statistic $Z_{Q,l}$. By neglecting any averaged double-frequency terms [4], these decision statistics are given for $A, B \in \{I, Q\}$ by

$$Z_{B,l} = E_c T \sqrt{\frac{P}{2}} \tilde{b}_{B,l}^{(1)} + N_{B,l} + T \sqrt{\frac{P}{2}} \sum_{k=2}^{K} I_{B,k,l}, \tag{1}$$

where $I_{I,k,l} = \frac{1}{T}\left[B^{I,1}_{I,k,l}\cos\theta_k - B^{I,1}_{Q,k,l}\sin\theta_k\right]$, $I_{Q,k,l} = \frac{1}{T}\left[B^{Q,1}_{I,k,l}\sin\theta_k + B^{Q,1}_{Q,k,l}\cos\theta_k\right]$, $B^{A,1}_{B,k,l} = \tilde{b}^{(k)}_{B,l-1}R^{B,A}_{m^k_l,<l>}(\tau_k) + \tilde{b}^{(k)}_{B,l}\hat{R}^{B,A}_{n^k_l,<l>}(\tau_k)$, $R^{B,A}_{m^k_l,<l>}(\tau_k) = \int_0^{\tau_k}\alpha^{(k)}_{B,m^k_l}(t-\tau_k)\alpha^{(1)}_{A,<l>}(t)dt$, $\hat{R}^{B,A}_{n^k_l,<l>}(\tau_k) = \int_{\tau_k}^{T}\alpha^{(k)}_{B,n^k_l}(t-\tau_k)\alpha^{(1)}_{A,<l>}(t)dt$, $N_{B,l}$ is normally distributed with zero mean and variance $\frac{N_oTE_c}{4}$, $\theta_k = \phi_k - \omega_c\tau_k$ is the phase angle of the kth carrier, the index m^k_l of the sequence used by the kth user depends on the delay τ_k and the demodulation interval l, and $n^k_l = m^k_l + 1$ modulo V. For convenience, we use indices $l-1$ and l in $\tilde{b}^{(k)}_{B,l-1}$ and $\tilde{b}^{(k)}_{B,l}$ to indicate the two consecutive modulated amplitudes.

3 System Analysis

We begin our analysis by calculating the Euclidean distances from the received sequence r to the correct sequence p and an error sequence q with length L for the 1st receiver [1]. Without loss of generality for the QAM signal set, we assume q begins from the first demodulation interval, and define $r = \frac{1}{TE_c}\sqrt{\frac{2}{P}}\{Z_{I,l}, Z_{Q,l}\}^L_{l=1}$, $p = \{\tilde{b}^{(1)}_{I,l}, \tilde{b}^{(1)}_{Q,l}\}^L_{l=1}$, and $q = \{\bar{b}^{(1)}_{I,l}, \bar{b}^{(1)}_{Q,l}\}^L_{l=1}$. The squared Euclidean distances between p and q, between r and p, and between r and q are given by

$$d^2 = \sum_{l=1}^{L}\left[(\bar{b}^{(1)}_{I,l} - \tilde{b}^{(1)}_{I,l})^2 + (\bar{b}^{(1)}_{Q,l} - \tilde{b}^{(1)}_{Q,l})^2\right] \triangleq \sum_{l=1}^{L}(\triangle^I_l)^2 + (\triangle^Q_l)^2, \tag{2}$$

$d^2_{r,p}$, and $d^2_{r,q}$ respectively. The difference between $d^2_{r,p}$ and $d^2_{r,q}$ can be expressed as

$$\triangle M(q,p) = d^2 - \sum_{k=2}^{K} W_k - \hat{N}, \tag{3}$$

where $W_k = \frac{2}{E_c}\sum_{l=1}^{L}(\triangle^I_l I_{I,k,l} + \triangle^Q_l I_{Q,k,l})$, and $\hat{N}$ is a zero-mean Gaussian random variable with variance $4\tilde{M}d^2\frac{N_o}{E_{Q,s}}$. We define the error weight sequence to be $\{\triangle^I_l, \triangle^Q_l\}^L_{l=1}$. The total MAI is given by $W = \sum_{k=2}^{K} W_k$.

3.1 Characterization of the MAI for Nonrectangular Chip Waveforms

In this subsection, we characterize the MAI when nonrectangular chip waveforms are used in the CV scheme. Our method is based on [3] and [5]. By applying the techniques used in [3], we show in Appendix A that the characterized MAI [3] from the kth user $W^{M,m}_k$ for the PAM and QAM signal sets is given by

$$W^{M,m}_k = \left[A_1\left(1 - \frac{s^M_k}{T_M}\right) + A_2\left(\frac{s^M_k}{T_M}\right)\right]\cos\theta_k \tag{4}$$

and

$$W^{M,m}_k = \left[A_1\left(1 - \frac{s^M_k}{T_M}\right) + A_2\left(\frac{s^M_k}{T_M}\right)\right]\cos\theta_k + \left[B_1\left(1 - \frac{s^M_k}{T_M}\right) + B_2\left(\frac{s^M_k}{T_M}\right)\right]\sin\theta_k \tag{5}$$

respectively, where s^M_k, T_M, M, and m are defined in [3]. The random variable $\frac{s^M_k}{T_M}$ is uniformly distributed on $[0,1)$. Before doing this, we can transform the matrix Z

given in Appendix A into the standard forms defined in [7] to reduce the computational complexity. The total characterized MAI, $W^{M,m}$, is given by $W^{M,m} = \sum_{k=2}^{K} W_k^{M,m}$. Since $W_k^{M,m}$ is of the form (4) or (5), we can use the method in [3] or [5] to compute the cumulative distribution function (cdf) $F_k^{M,m}$ of $W_k^{M,m}$. Now we define the discrete characterizing MAI $D_k^{M,N_u,m}$, where N_u is a positive integer, as follows: $D_k^{M,N_u,m} = \frac{i}{N_u}$ if $W_k^{M,m} \in [\frac{i-\frac{1}{2}}{N_u}, \frac{i+\frac{1}{2}}{N_u}]$, where i is an integer. The probability mass function (pmf) for $D_k^{M,N_u,m}$ is given by $d_k^{M,N_u,m}(i) = F_k^{M,m}(\frac{i+\frac{1}{2}}{N_u}) - F_k^{M,m}(\frac{i-\frac{1}{2}}{N_u})$. We define the total characterizing MAI $D^{M,N_u,m}$ by $D^{M,N_u,m} = \sum_{k=2}^{K} D_k^{M,N_u,m}$. Since the random variables $D_k^{M,N_u,m}$, $2 \leq k \leq K$, form a set of independent random variables, the pmf of $D^{M,N_u,m}$ is then given by $d^{M,N_u,m} = d_2^{M,N_u,m} * d_3^{M,N_u,m} * \cdots * d_K^{M,N_u,m}$, where the asterisk denotes a convolution.

3.2 Effect of Using Cascaded-Sequence Schemes on PEP

In the remaining part of this paper, we neglect the thermal noise, and focus on the MAI part. The PEP is then given by

$$\Pr(p \longrightarrow q) = \Pr(W \geq d^2). \tag{6}$$

The numerical examples in this paper are based on the following assumptions. For the PAM signal set, we compare the uncoded BPSK and TCM/4-PAM systems, both using DS/SSMA and various cascaded-sequence schemes. The encoder is a rate 1/2, 4-state convolutional encoder [10]. The spreading sequences are Gold sequences [9] with length $N = 31$. For computational simplicity, we use rectangular-chip waveforms. We select 95 Gold sequences and assign them to 19 users. Each user has a set of 5 spreading sequences. We consider four cascaded-sequence schemes: C1, C3, C4, and C5. For example, in the C4 scheme, each user uses its first 4 spreading sequences periodically. We assume the all-zeros path to be the correct path and consider the first four error paths ($X^iY^jZ^k$): 1st path $X^{36}Y^3Z^2$, 2nd path $X^{40}Y^4Z^4$, 3rd path $X^{40}Y^5Z^2$, and 4th path $X^{44}Y^5Z^4$, where the exponents i = squared Euclidean distance from the error path to the correct path, j = length of the error path, and k = number of error bits corresponding to the error path [8]. The corresponding error sequences are given by q_1, q_2, q_3, and q_4. For the QAM signal set, we consider the TCM/16-QAM system. We select 96 Gold sequences and assign them to 16 users. We use the same encoder and assume the all-zeros path to be the correct path. We only give an approximation of the variance in the C3 scheme. The error paths we consider are $X^{16}Y^1Z^1$, $X^{16}Y^1Z^2$, and$X^{20}Y^3Z^2$. The corresponding error sequences are given by q_1, q_2, and q_3. We give the key proposition first.

Proposition : In a CV scheme, the variance of the characterized MAI from the kth user on an error sequence with squared Euclidean distance d^2 from the correct sequence is approximately equal to $\mathrm{Var}(W_k^{M,m}) \approx 4d^2(2\tilde{M})\tilde{C}_{Q,V}^k$ for the QAM signal set, and $\mathrm{Var}(W_k^{M,m}) \approx 4d^2\tilde{M}\tilde{C}_{P,V}^k$ for the PAM signal set, where $\tilde{C}_{Q,V}^k$ and $\tilde{C}_{P,V}^k$ are defined in Appendix B. *Proof:* See Appendix B.

Since the random variables $W_k^{M,m}$ are independent, the variance of the total characterized MAI $W^{M,m}$ is given by $\mathrm{Var}(W^{M,m}) \approx 4d^2(2\tilde{M})\sum_{k=2}^{K}\tilde{C}_{Q,V}^k \triangleq 4d^2(2\tilde{M})\tilde{C}_{Q,V}$ for the QAM signal set, and $\mathrm{Var}(W^{M,m}) \approx 4d^2\tilde{M}\sum_{k=2}^{K}\tilde{C}_{P,V}^k \triangleq 4d^2\tilde{M}\tilde{C}_{P,V}$ for the PAM signal set. The results in Fig. 3 and Fig. 4 show that the above approximation is very accurate. We see that the MAI variance depends mainly on d^2, but not on the length of the corresponding error path. In Fig. 5, we see that the cascaded-sequence schemes can greatly reduce the PEP until the length of the cascaded sequence is greater than that of the error path. On the other hand, the error performance of the uncoded system is affected very little by using a cascaded-sequence scheme instead of the single-sequence scheme. This can be explained as follows. From Fig. 6, we see that, for the spreading sequences used in this paper, the values of $\tilde{C}_{P,V}$ are approximately the same when different CV schemes are used. This means that the MAI variance from a particular error path is approximately the same for any CV scheme we use. In Fig. 7, we see that the MAI contributions from different interfering users in the 2nd error path cancel each other more often in the C4 and C5 schemes than in the C1 and C3 schemes, where the threshold d^2 in (6) equals 40. This is illustrated by the matrix $ma_i^{a,b}$ defined in Appendix A. In Fig. 8, we see that, although the PEP's are monotonically decreasing in d^2, the PEP's due to q_2, q_3, and q_4 are not negligible even in the C5 scheme.

3.3 Analytical Performance Bound

It is known that the MAI will tend to a Gaussian random variable when the number of simultaneous users K becomes large [6]. We derive the results for the PAM signal set only. If we assume that the MAI from each interfering user is Gaussian distributed, we can use the methods in [8] and [11] to derive the Gaussian bound (GB) given by

$$GB = \frac{1}{n}\left[Q\left(\sqrt{\frac{d^2_{nf}}{4\tilde{C}_{P,V}}}\right)\exp\left(\frac{d^2_{nf}}{8\tilde{C}_{P,V}}\right)\right] \times \frac{\partial T(X,Y,Z)}{\partial Z} \tag{7}$$

where $X = \exp(-\frac{1}{8\tilde{C}_{P,V}})$, $Y = 2^{1-(n+1)}$, $Z = 1$, $d^2_{nf} = d^2_f/\tilde{M}$ is the normalized free squared Euclidean distance, $T(X,Y,Z) = \sum_{i,j,k} A_{i,j,k} X^i Y^j Z^k$ is the modified generating function and $A_{i,j,k}$ is the number of error paths characterized by the exponents i, j, and k. The exponent i represents the normalized squared Euclidean distance here.

From the matrix expression of the MAI, we see that the pdf's of the MAI depend on the error weight sequences. That is, the PEP's may not be the same for different pairs of correct and incorrect paths with the same distance in the trellis. Our approach to get the UUB on the BEP is as follows. We calculate the exact PEP's for the first v error paths with respect to a specific correct path in a CV scheme, where V is an integer greater than or equal to the length of the 1st error path, and v is the number of error paths with length less than or equal to V. We assume that the PEP's are the same for the error paths specified by the same exponents i, j, and k as those of the first v error paths, and replace these PEP's in (7) by the first v PEP's. We call the altered part in (7) the lower bound (LB), and the remaining unaltered part the exponential bound (EB). The upper bound UB on the UUB is given by $UB = LB + EB$.

In Fig. 9, we see that when K is small or moderate, the UB converges very quickly, and the TCM system is better than the uncoded system. When K is large, the GB and UB merge, since the MAI tends to be a Gaussian random variable. At this point, the UUB itself is very loose, and another method is needed to bound the error probability of the TCM system due to the MAI.

Appendix A

In this appendix, we derive an expression of the MAI in matrix form when the cascaded-sequence schemes are used. We consider the C3 scheme. The case of the general CV scheme follows in the same manner.

We consider the MAI from the kth user, and define $C^{a,b}_{n,m}(\cdot)$, where $a, b \in \{I, Q\}$, as the discrete aperiodic cross-correlation function [2] of the nth spreading sequence used by the kth user in the a-channel and the mth spreading sequence used by the 1st user in the b-channel. We also define the row vector $C^{a,b}_{n,m}(x:y) = [C^{a,b}_{n,m}(x), C^{a,b}_{n,m}(x+1), \cdots, C^{a,b}_{n,m}(y)]$. For the PAM signal set, we can express the MAI from the kth user W_k as

$$\begin{array}{l}\frac{2}{E_c}\sum\limits_{l=1}^{L}\triangle_l\{\tilde{b}^{(k)}_{l-1}[C^{I,I}_{m^k_l,<l>}(\lambda_k - N)\hat{R}_\psi(\tau_k - \lambda_k T_c) + C^{I,I}_{m^k_l,<l>}(\lambda_k - N + 1)R_\psi(\tau_k - \lambda_k T_c)] \\ +\tilde{b}^k_l[C^{I,I}_{n^k_l,<l>}(\lambda_k)\hat{R}_\psi(\tau_k - \lambda_k T_c) + C^{I,I}_{n^k_l,<l>}(\lambda_k + 1)R_\psi(\tau_k - \lambda_k T_c)]\} \times \cos\theta_k,\end{array} \tag{8}$$

where $\hat{R}_\psi$, R_ψ, and λ_k are defined in [4]. By replacing $\hat{R}_\psi$ by $\hat{R}^{M,m}$, R_ψ by $R^{M,m}$, and E_c by $E^{M,m}_c$ in (8), we obtain the characterized MAI from the kth user given by (4), where $\hat{R}^{M,m}$, $R^{M,m}$, and $E^{M,m}_c$ are defined in [3]. The equation (5) can be obtained in a similar way for QAM signal sets. In the C3 scheme, we have to consider three different demodulation intervals. We give expressions of A_1, A_2, B_1, and B_2 for the error sequence with length 3 when the QAM signal set is used. For notational simplicity, we omit the superscripts M and m in the parameters $\hat{R}^{M,m}$ and $R^{M,m}$, and the superscript (k) in the modulated amplitudes $\{\tilde{b}^{(k)}_{I,l}, \tilde{b}^{(k)}_{Q,l}\}$ in this Appendix.

We define the matrices $ma^{a,b}_1$, $ma^{a,b}_2$, $we^{a,b}_i$, where $i \in \{1, \ldots, V\} = \{1, 2, 3\}$, as

$$ma^{a,b}_1 = \begin{bmatrix} C^{a,b}_{3,1}(-N:-1) & C^{a,b}_{2,1}(-N:-1) & C^{a,b}_{1,1}(-N:-1) \\ C^{a,b}_{1,1}(0:N-1) & C^{a,b}_{3,1}(0:N-1) & C^{a,b}_{2,1}(0:N-1) \\ C^{a,b}_{1,2}(-N:-1) & C^{a,b}_{3,2}(-N:-1) & C^{a,b}_{2,2}(-N:-1) \\ C^{a,b}_{2,2}(0:N-1) & C^{a,b}_{1,2}(0:N-1) & C^{a,b}_{3,2}(0:N-1) \\ C^{a,b}_{2,3}(-N:-1) & C^{a,b}_{1,3}(-N:-1) & C^{a,b}_{3,3}(-N:-1) \\ C^{a,b}_{3,3}(0:N-1) & C^{a,b}_{2,3}(0:N-1) & C^{a,b}_{1,3}(0:N-1) \end{bmatrix}, \tag{9}$$

$$ma^{a,b}_2 = \begin{bmatrix} C^{a,b}_{3,1}(-N+1:0) & C^{a,b}_{2,1}(-N+1:0) & C^{a,b}_{1,1}(-N+1:0) \\ C^{a,b}_{1,1}(1:N) & C^{a,b}_{3,1}(1:N) & C^{a,b}_{2,1}(1:N) \\ C^{a,b}_{1,2}(-N+1:0) & C^{a,b}_{3,2}(-N+1:0) & C^{a,b}_{2,2}(-N+1:0) \\ C^{a,b}_{2,2}(1:N) & C^{a,b}_{1,2}(1:N) & C^{a,b}_{3,2}(1:N) \\ C^{a,b}_{2,3}(-N+1:0) & C^{a,b}_{1,3}(-N+1:0) & C^{a,b}_{3,3}(-N+1:0) \\ C^{a,b}_{3,3}(1:N) & C^{a,b}_{2,3}(1:N) & C^{a,b}_{1,3}(1:N) \end{bmatrix}, \tag{10}$$

$$we^{a,b}_1 = [\tilde{b}_{a,0}\Delta^b_1, \tilde{b}_{a,1}\Delta^b_1, \tilde{b}_{a,1}\Delta^b_2, \tilde{b}_{a,2}\Delta^b_2, \tilde{b}_{a,2}\Delta^b_3, \tilde{b}_{a,3}\Delta^b_3], \tag{11}$$

$$we^{a,b}_2 = [\tilde{b}_{a,2}\Delta^b_3, \tilde{b}_{a,3}\Delta^b_3, \tilde{b}_{a,0}\Delta^b_1, \tilde{b}_{a,1}\Delta^b_1, \tilde{b}_{a,1}\Delta^b_2, \tilde{b}_{a,2}\Delta^b_2], \tag{12}$$

and

$$we^{a,b}_3 = [\tilde{b}_{a,1}\Delta^b_2, \tilde{b}_{a,2}\Delta^b_2, \tilde{b}_{a,2}\Delta^b_3, \tilde{b}_{a,3}\Delta^b_3, \tilde{b}_{a,0}\Delta^b_1, \tilde{b}_{a,1}\Delta^b_1]. \tag{13}$$

Note that the dimension of the matrix $ma^{a,b}_i$ is $2V \times NV = 6 \times 3N$, and the dimension of the matrix $we^{a,b}_i$ is $1 \times 2V = 1 \times 6$.

For a fixed sequence of modulated amplitudes $\{\tilde{b}_{I,l}, \tilde{b}_{Q,l}\}^3_{l=0}$ of the kth user, a fixed value $\lambda \in \{0, 1, \ldots, M-1\}$, and a fixed value $i \in \{1, 2, \ldots, V\} = \{1, 2, 3\}$, if we put $Z = [A_1, A_2, B_1, B_2]^T$, then Z can be determined by taking each column vector of the $4 \times VN = 4 \times 3N$ matrix $\bar{Z}$ given by $\bar{Z} = [\bar{A}_1, \bar{A}_2, \bar{B}_1, \bar{B}_2]^T$, where,

$$\begin{aligned} \bar{A}_1 &= \tfrac{2}{N} \times we^{I,I}_i \times \left[\tfrac{\hat{R}(\lambda)}{ME^{M,m}_c} ma^{I,I}_1 + \tfrac{R(\lambda)}{ME^{M,m}_c} ma^{I,I}_2\right] \\ &+ \tfrac{2}{N} \times we^{Q,Q}_i \times \left[\tfrac{\hat{R}(\lambda)}{ME^{M,m}_c} ma^{Q,Q}_1 + \tfrac{R(\lambda)}{ME^{M,m}_c} ma^{Q,Q}_2\right], \end{aligned} \tag{14}$$

$$\begin{aligned}\bar{A}_2 &= \tfrac{2}{N} \times we_i^{I,I} \times \left[\tfrac{\hat{R}(\lambda+1)}{ME_c^{M,m}} ma_1^{I,I} + \tfrac{R(\lambda+1)}{ME_c^{M,m}} ma_2^{I,I}\right] \\ &+ \tfrac{2}{N} \times we_i^{Q,Q} \times \left[\tfrac{\hat{R}(\lambda+1)}{ME_c^{M,m}} ma_1^{Q,Q} + \tfrac{R(\lambda+1)}{ME_c^{M,m}} ma_2^{Q,Q}\right],\end{aligned} \tag{15}$$

$$\begin{aligned}\bar{B}_1 &= -\tfrac{2}{N} \times we_i^{Q,I} \times \left[\tfrac{\hat{R}(\lambda)}{ME_c^{M,m}} ma_1^{Q,I} + \tfrac{R(\lambda)}{ME_c^{M,m}} ma_2^{Q,I}\right] \\ &+ \tfrac{2}{N} \times we_i^{I,Q} \times \left[\tfrac{\hat{R}(\lambda)}{ME_c^{M,m}} ma_1^{I,Q} + \tfrac{R(\lambda)}{ME_c^{M,m}} ma_2^{I,Q}\right],\end{aligned} \tag{16}$$

and

$$\begin{aligned}\bar{B}_2 &= -\tfrac{2}{N} \times we_i^{Q,I} \times \left[\tfrac{\hat{R}(\lambda+1)}{ME_c^{M,m}} ma_1^{Q,I} + \tfrac{R(\lambda+1)}{ME_c^{M,m}} ma_2^{Q,I}\right] \\ &+ \tfrac{2}{N} \times we_i^{I,Q} \times \left[\tfrac{\hat{R}(\lambda+1)}{ME_c^{M,m}} ma_1^{I,Q} + \tfrac{R(\lambda+1)}{ME_c^{M,m}} ma_2^{I,Q}\right].\end{aligned} \tag{17}$$

Appendix B

We discuss the QAM signal set first. From the results of Appendix A, we have, since $W_k^{M,m} = (A_1(1-U_k)+A_2U_k)\cos\theta_k + (B_1(1-U_k)+B_2U_k)\sin\theta_k$, where U_k and θ_k are independent random variables and are uniformly distributed on $[0,1)$ and $[0,2\pi)$ respectively, $E(W_k^{M,m}) = 0$ and $\mathrm{Var}(W_k^{M,m}|A_1, A_2, B_1, B_2) = \frac{1}{6}(A_1^2 + A_1A_2 + A_2^2 + B_1^2 + B_1B_2 + B_2^2)$. For notational simplicity, we omit the superscripts M and m in the parameters $\hat{R}^{M,m}$ and $R^{M,m}$ in this appendix. We define the parameters S_1, S_2, $\mu_{n,m}^{a,b}(i)$, $\mu_V^{a,b}(0)$ and $\mu_V^{a,b}(1)$ by $S_1 = (\frac{1}{3M})(\frac{1}{ME_c^{M,m}})^2\{\sum_{\lambda=0}^{M-1}[(R(\lambda))^2 + (R(\lambda+1))^2 + R(\lambda)R(\lambda+1)]\}$, $S_2 = (\frac{1}{6M})(\frac{1}{ME_c^{M,m}})^2 \times \{\sum_{\lambda=0}^{M-1}[2\hat{R}(\lambda)R(\lambda) + 2\hat{R}(\lambda+1)R(\lambda+1) + \hat{R}(\lambda)R(\lambda+1) + \hat{R}(\lambda+1)R(\lambda)]\}$, $\mu_{n,m}^{a,b}(i) = \sum_{l=1-N}^{N-1} C_{n,m}^{a,b}(l)C_{n,m}^{a,b}(l+i)$, $\mu_V^{a,b}(0) = (\frac{1}{V^2N^3})(\sum_{n=1}^{V}\sum_{m=1}^{V}\mu_{n,m}^{a,b}(0))$, and $\mu_V^{a,b}(1) = (\frac{1}{V^2N^3})(\sum_{n=1}^{V}\sum_{m=1}^{V}\mu_{n,m}^{a,b}(1))$. Note that the values of S_1 and S_2 depend on the chip waveform used, and the values of $\mu_V^{a,b}(0)$ and $\mu_V^{a,b}(1)$ depend on the spreading sequences used. For a rectangular chip waveform, $S_1 = \frac{1}{3}$ and $S_2 = \frac{1}{6}$. These normalized parameters are useful in the optimization of the system performance. To get the unconditional variance $\mathrm{Var}(W_k^{M,m})$, we take the expectation values with respect to A_1, A_2, B_1, and B_2. Since some of the cross-product terms are small, we may neglect them and have the approximation:

$$\mathrm{Var}(W_k^{M,m}) \approx 4d^2(2\tilde{M})(\mu_V(0)S_1 + \mu_V(1)S_2) \triangleq 4d^2(2\tilde{M})\tilde{C}_{Q,V}^k, \tag{18}$$

where $\mu_V(i) = \frac{1}{4}(\mu_V^{I,I}(i) + \mu_V^{Q,Q}(i) + \mu_V^{I,Q}(i) + \mu_V^{Q,I}(i))$ for $i = 0, 1$. If we set $V = 1$, $\tilde{M} = 1$, $\triangle_l^I = 0$ or $\triangle_l^Q = 0$, and scale (18) by 4, we have the same result as (3.12) in [4]. For the PAM signal set, we have

$$\mathrm{Var}(W_k^{M,m}) \approx 4d^2\tilde{M}(\mu_V^{I,I}(0)S_1 + \mu_V^{I,I}(1)S_2) \triangleq 4d^2\tilde{M}\tilde{C}_{P,V}^k. \tag{19}$$

If we set $V = 1$, $\tilde{M} = 1$, $d^2 = 1$, and scale (19) by 4, we have the same results as those in [2] and [4]. It is worthwhile to point out that $(T_c)^3(E_c^{M,m})^2S_1$ and $(T_c)^3(E_c^{M,m})^2S_2$ are the characterizations of the parameters $\int_0^{T_c} R_\psi^2(\tau)d\tau$ and $\int_0^{T_c} R_\psi(\tau)\hat{R}_\psi(\tau)d\tau$ defined in [4] respectively.

References

[1] H. Chu, "Trellis-coded modulation for CDMA," Ph.D. thesis (Advisor: J. S. Lehnert) , Purdue University, West Lafayette, Indiana, May 1994.

[2] M. B. Pursley, "Performance evaluation for phase-coded spread-spectrum multiple-access communication - Part I: System analysis," *IEEE Trans. Commun.*, vol. COM-25, pp. 795-799, Aug. 1977.

[3] R. T. Hsu and J. S. Lehnert, "A characterization of multiple-access interference in generalized quadriphase spread-spectrum communications," *IEEE Trans. Commun.*, vol. COM-42, pp. 2001-2010, Feb. 1994.

[4] M. B. Pursley, F. D. Garber, and J. S. Lehnert, "Analysis of generalized quadriphase spread-spectrum communications," *in Conf. Rec., IEEE Int. Conf. Commun.*, June 1980, vol. 1, pp. 15.3.1-15.3.6.

[5] J. S. Lehnert, "An efficient technique for evaluating direct-sequence spread-spectrum multiple-access communications,"*IEEE Trans. Commun.*, vol. COM-37, pp. 851-858, Aug. 1989.

[6] R. K. Morrow, Jr., and J. S. Lehnert, "Bit-to-bit error dependence in slotted DS/SSMA packet systems with random signature sequences,"*IEEE Trans. Commun.*, vol. COM-37, pp. 1052-1061, Oct. 1989.

[7] A. T. McDowell, "Dual-phase continuous phase modulation for bandwidth efficient spread-spectrum multiple-access communication", Ph.D. thesis (Advisor: J. S. Lehnert) , Purdue University, West Lafayette, Indiana, May 1993.

[8] E. Zehavi and J. K. Wolf, "On the performance evaluation of trellis codes,"*IEEE Trans. Inform. Theory.*, vol. IT-33, pp. 196-202, Mar. 1987.

[9] M. B. Pursley and H. F. A. Roefs, "Numerical evaluation of correlation parameters for optimal phases of binary shift-register sequences,"*IEEE Trans. Commun.*, vol. COM-27, pp. 1597-1604, Oct. 1979.

[10] G. Ungerboeck, "Channel coding with multilevel/phase signals,"*IEEE Trans. Inform. Theory*, vol. IT-30, no. 1, pp. 55-67, Jan. 1982.

[11] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*, New York: McGraw-Hill, 1979.

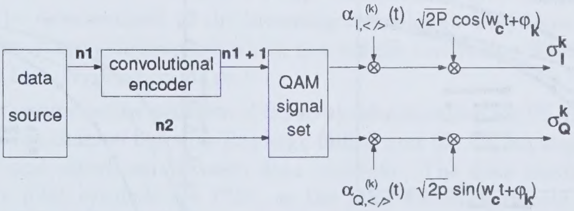

$\alpha^{(k)}_{I,</>}(t)$
$\sqrt{2P}\cos(w_c t+\varphi_k)$
data
source
n1
convolutional encoder
n1 + 1
QAM signal set
n2
σ^k_I
σ^k_Q
$\alpha^{(k)}_{Q,</>}(t)$
$\sqrt{2p}\sin(w_c t+\varphi_k)$

Figure 1: Transmitter for the *k*th user.

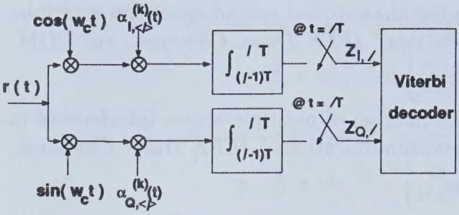


Figure 2. Correlation receiver for the kth user.

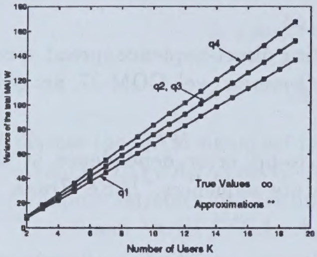


Figure 3. Exact values vs. approximations of the variances of the total MAI $W=\sum_{k=2}^{K} W_k$ from the first 4 error paths when C5 scheme is used in TCM/4-PAM.

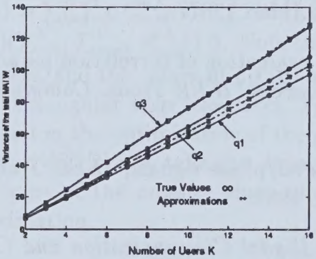


Figure 4. Exact values vs. approximations of the variances of the total MAI $W=\sum_{k=2}^{K} W_k$ from the first 3 error paths when C3 scheme is used in TCM/16-QAM.

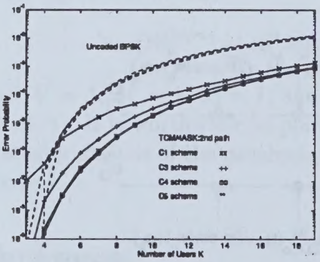


Figure 5. Uncoded BPSK (BEP) vs. TCM/4-PAM (PEP of 2nd error path: $X^{40}Y^4Z^4$) when C1, C3, C4, and C5 schemes are used.

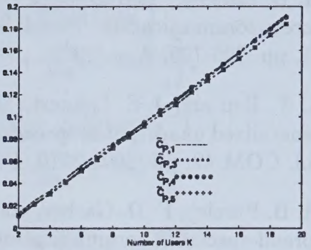


Figure 6. Values of the parameter $\tilde{C}_{P,V}$ when C1, C3, C4, and C5 schemes are used in TCM/4-PAM.

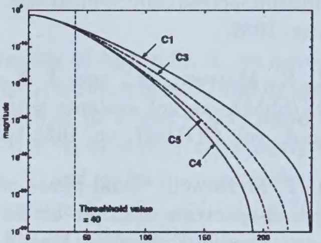


Figure 7. Probability density functions of the MAI on the 2nd error path ($X^{40}Y^4Z^4$) when C1, C3, C4, and C5 schemes are used in TCM/4-PAM.

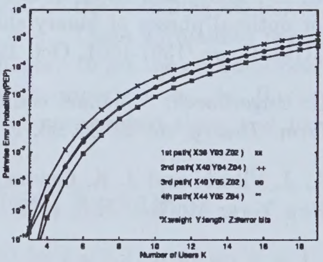


Figure 8. The monotonic property of PEP's in d^2 when C5 scheme is used in TCM/4-PAM.

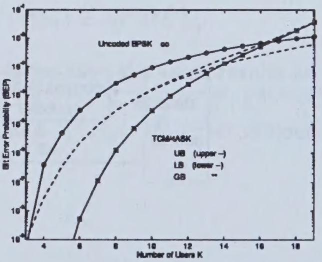


Figure 9. Uncoded BPSK v.s. TCM/4-PAM when C5 scheme is used.

Frame Synchronization for Pilot Symbol Assisted Modulation*

J. A. Gansman, M. P. Fitz, and J. V. Krogmeier
School of Electrical Engineering, Purdue University, W. Lafayette, IN

Abstract

Pilot symbol assisted modulation (PSAM) is a method to reduce the effects of fading in mobile communications by periodically inserting known symbols in the data stream. The receiver uses these pilot symbols to derive its amplitude and phase reference. One aspect of this procedure which has not received much attention in the literature is the method used by the receiver to locate the pilot symbols. This paper satisfies that need by investigating optimum and sub-optimum frame synchronization techniques for the case of an M–QAM data symbol set and a PSK pilot sequence.

1 Introduction

One method to reduce the effects of fading in mobile communications is to periodically insert known symbols in the data stream which are used by the receiver to derive its amplitude and phase reference. This is referred to as pilot symbol assisted modulation (PSAM) [1]. In practice a frame of N pilot symbols is periodically repeated. This structure eliminates undesired in-band tones and yet synchronization is easily obtained. One aspect of PSAM based demodulation often omitted in the literature is the method used to obtain frame synchronization.

Previous work on frame synchronization either concentrated on AWGN channels [2, 3] or used channel state estimates for synchronization [4]. Neither of these approaches is appropriate for PSAM on a fading channel. Furthermore, this prior work selected the frame sequence from the same symbol set as the data, so the performance was limited by the probability that the known sequence appears as part of the random data. A similar problem occurs in symbol time estimation for spread spectrum systems where a reference codeword has to be synchronized to the incoming data signal. Polydoros and Weber [5] characterized a two stage synchronizer which is a simple correlation filter and threshold detector followed by a verification stage.

This paper examines the problem of frame synchronization for PSAM which differs from [2, 3] in that the channel has slow Rayleigh fading and the known sequence is spread across the frame and interlaced between data symbols. The data modulation used is M–QAM and the pilot symbols are PSK, so the pilot sequence cannot appear in the random data. This removes the upper limit on the probability of correct acquisition. Only the statistical characteristics of the fading are assumed to be known. The optimum

---

*This work was supported by NSF under contract 9406073, the Dept. of Transportation under grant DTFH61-92-X-0026, and by an AFCEA fellowship.

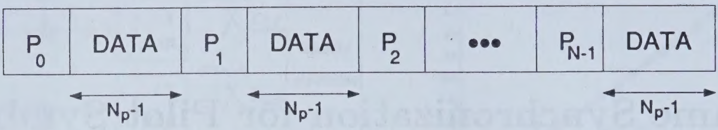


Figure 1: PSAM Frame Structure

synchronizer turns out to be computationally prohibitive, so the two stage system of [5] is examined. The first stage is a simple correlation filter and threshold detector. The second stage computes the correlation over A consecutive frames. If at least B of these exceed the threshold, synchronization is achieved.

2 Data Model

The data symbols, d_k, are selected from an alphabet, Ω_d, of size M (16 QAM for this paper) and normalized so that $\mathrm{E}\left\{ \mid d_k \mid^2 \right\} = 1$. The pilot symbols, p_k, are selected from an alphabet, Ω_p, (8 PSK for this paper) and normalized so that $\mathrm{E}\left\{ \mid p_k \mid^2 \right\} = 1$. The pilots are inserted after every $N_p - 1$ data symbols to form a frame of length $L = NN_p$ symbols (see Figure 1) which is linearly modulated using a square-root Nyquist pulse with unit energy. The received signal is passed through a filter matched to the pulse shape and sampled at the correct symbol time so no ISI is present. The three channel models considered, are AWGN with unknown phase, AWGN with a small frequency offset, and frequency nonselective slow Rayleigh fading. The matched filter outputs are modeled as

$$x(k) = \sqrt{E_s} e^{j\theta} y(k) + n(k) \tag{1}$$

$$x(k) = \sqrt{E_s} e^{j(2\pi f_o k + \theta)} y(k) + n(k) \tag{2}$$

$$x(k) = \sqrt{E_s} c(k) y(k) + n(k) \tag{3}$$

where $y(k)$ is a pilot or data symbol, θ is a phase offset uniformly distributed over $[-\pi, \pi]$, $n(k)$ is zero mean AWGN with variance N_o, f_o is a frequency offset uniformly distributed over $[-f_m, f_m]$, and $c(k)$ is zero mean complex Gaussian multiplicative distortion with variance σ_c^2. The fading process has the U-shaped spectrum of isotropic scattering

$$\mathrm{E}\{ c(n) c^*(n-k) \} = \sigma_c^2 J_0(2\pi k f_D), \tag{4}$$

where $J_0(\bullet)$ is the zero-th order Bessel function, $f_D = F_D/F_s$ is the Doppler spread normalized by the symbol rate, and the slow fading assumption implies that $f_D \ll 1$. This is simulated using Jakes Model [6].

3 Synchronization in AWGN

3.1 ML Estimation

Consider a fixed observation vector of length $L = NN_p$ at some time instant n_o,

$$\mathrm{x} = [x(n_o), x(n_o+1), \ldots, x(n_o+L-1)]^T, \tag{5}$$

and let $x_k = x(n_o + k)$ for $0 \leq k \leq L-1$. The frame synchronization problem is to estimate the index J which corresponds to the start of a superframe. Since no prior information is known about the start of the frame, Maximum Likelihood (ML) estimation is the optimum estimator of J. The ML estimate is given by

$$\hat{J}_{ML} = \underset{J\in[0,L-1]}{\text{Argmax}} \; f(\mathbf{x}/J) \tag{6}$$

where $f(\mathbf{x})$ is the probability density function (pdf) of $\mathbf{x}$. In the AWGN case (1), the observations, x_k, are i.i.d. and conditionally Gaussian, so $f(\mathbf{x}/J)$ can be written as (see Appendix A)

$$f(\mathbf{x}/J) = \int_{-\pi}^{\pi} \frac{C \, \exp\left(\frac{2\sqrt{E_s}}{N_o} \mathcal{R}e\left\{\sum_{l=0}^{N-1} x_{lN_p+J} p_l^* e^{-j\theta}\right\}\right)}{\sum_{\mathbf{d}\in\Omega_d^N} \exp\left(-\frac{E_s\|\mathbf{d}\|^2}{N_o}\right) \exp\left(\frac{2\sqrt{E_s}}{N_o} \mathcal{R}e\left\{\sum_{l=0}^{N-1} x_{lN_p+J} d^* e^{-j\theta}\right\}\right)} d\theta \tag{7}$$

where shifts in the subscripts are modulo–L. No closed-form solution exists for (7), but it shows that the sufficient statistics are the observations $x(n + lN_p)$ for $l = 0, 1, \ldots$.

3.2 Hypothesis Testing

Since the ML estimate is impractical to implement, view the synchronization problem as deciding if $x(n) = p_0$, based on the L observations $\{x(n), x(n+1), \ldots, x(n+L-1)\}$. The preceding analysis showed that only $\{x(n), x(n+Np), \ldots, x(n+(N-1)N_p)\}$ need to be considered. Therefore, the L-length observation vector $\mathbf{x}(n)$ is replaced with the following N-length vector of observations at the proposed pilot times

$$\mathbf{z}(n) = [\, x(n), \; x(n+N_p), \; \ldots, \; x(n+(N-1)N_p)\,]^T. \tag{8}$$

Each observation symbol $x(n)$ has the following three possible hypotheses

$$\begin{array}{lll} \mathrm{H}_0: & y(n) \text{ is a data symbol} & \mathbf{z}(n) = \sqrt{E_s}\,\mathbf{d}e^{j\theta} + \mathbf{n} \\ \mathrm{H}_1: & y(n) \text{ is } p_0 & \mathbf{z}(n) = \sqrt{E_s}\,\mathbf{p}e^{j\theta} + \mathbf{n} \\ \mathrm{H}_2: & y(n) \text{ is } p_k \text{ for } 1 \leq k \leq N-1 & \mathbf{z}(n) = \sqrt{E_s}\,\mathbf{T}^{(k)H}\mathbf{p}e^{j\theta} + \mathbf{n} \end{array}$$

where $\mathbf{T}$ is a rotation matrix. The probability of false acquisition due to random data dominates the performance, so H_2 will be ignored and the hypothesis test compares the likelihood ratio between H_0 and H_1 to a threshold. Proceeding as before yields

$$L(n) = \frac{\left(Me^{-E_s/N_o}\right)^N I_0\left(\frac{2\sqrt{E_s}}{N_o}\left|\sum_{k=0}^{N-1} x(n+kN_p)p_k^*\right|\right)}{\sum_{\mathbf{d}\in\Omega_d^N} \exp\left(-\frac{E_s\|\mathbf{d}\|^2}{N_o}\right) I_0\left(\frac{2\sqrt{E_s}}{N_o}\left|\sum_{k=0}^{N-1} x(n+kN_p)d_k^*\right|\right)}. \tag{9}$$

Note that this is equivalent to integrating the numerator and denominator of (7) separately, and the minimum probability of error detector is achieved with a threshold of 1. Taking the log and applying a high SNR approximation reduces the hypothesis test to

$$\left|\frac{1}{N}\sum_{k=0}^{N-1} x(n+kN_p)p_k^*\right|^2 - \frac{N_o}{2N\sqrt{E_s}} g(n) \underset{\mathrm{H}_0}{\overset{\mathrm{H}_1}{\gtrless}} \tfrac{1}{2}\sqrt{E_s}\left[1 - \frac{\ln M}{E_s/N_o}\right] \tag{10}$$

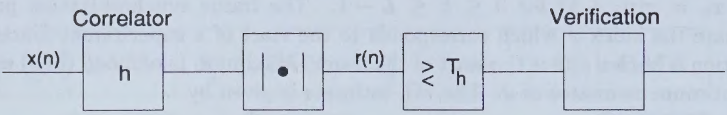


Figure 2: Two stage correlation detector

$$g(n) = \ln\left[\sum_{\mathbf{d}\in\Omega_d^N} \exp\left\{\frac{2\sqrt{E_s}}{N_o}\left|\sum_{k=0}^{N-1} x(n+kN_p)d_k^*\right| - \frac{E_s\|\mathbf{d}\|^2}{N_o}\right\}\right]. \tag{11}$$

Equation (10) is a correlator and a data correction term which averages over all possible N-length data vectors. The size of the data sets makes the computation of $g(n)$ prohibitive. Therefore, it is ignored and the two stage correlator [5] shown in Figure 2 is used. The received sequence is passed through a filter with impulse response

$$h(n) = \begin{cases} \frac{1}{N}p^*_{N-1-l} & n = lN_p \quad 0 \le l \le N-1 \\ 0 & \text{else} \end{cases} \tag{12}$$

and the magnitude of the output is compared to a threshold T_h. When the threshold is exceeded, a verification stage computes the correlation over the next A consecutive frames. If at least B of these exceed the threshold, then synchronization is achieved; otherwise the first stage continues. Note that the synchronizer need only compute one correlation in each symbol duration and store the results of the last $(A+1)L$ threshold comparisons.

4 Sync. in AWGN with a Frequency Offset

Since the pilot symbols are used to estimate the channel states, it is unnecessary to perform frequency estimation. Therefore, consider an unknown frequency offset f_o between the transmitter and receiver (2). Ignoring data dependent terms as before, the threshold detector is

$$\frac{1}{2f_m}\int_{-f_m}^{f_m} I_0\left(\frac{2\sqrt{E_s}}{N_o}\left|\sum_{k=0}^{N-1} x(n+kN_p)p_k^* e^{-j2\pi f_o N_p k}\right|\right) df_o \underset{\mathrm{H_0}}{\overset{\mathrm{H_1}}{\gtrless}} T_h. \tag{13}$$

No closed-form solution exists for this integral. However, using $I_0(x) \approx |x|^2$, and normalizing by the received energy (to make threshold selection easier) yields a good threshold test. Defining s_k as the de-rotated observation symbol (i.e., $s_k = x(n+kN_p)p_k^*$) yields

$$r(n) = \sum_{m=-(N-1)}^{N-1} \phi(m)\,\mathrm{sinc}(2\pi f_m N_p m). = \frac{\mathbf{s}^H(n)\mathbf{R}\mathbf{s}(n)}{\mathbf{s}^H(n)\mathbf{s}(n)} \underset{\mathrm{H_0}}{\overset{\mathrm{H_1}}{\gtrless}} T_h \tag{14}$$

where $\mathbf{R}_{k,l} = \mathrm{sinc}[2\pi f_m N_p(k-l)]$ and $\phi(m) = \sum_{l=L_1}^{L_2} s^*(n+(m+l)N_p)s(n+lN_p)/\|\mathbf{s}\|^2$ with $L_1 = \max(0,-m)$ and $L_2 = \min(N-1, N-1-m)$. Note that $\phi(m)$ is the sample autocorrelation of s_k. Consequently, designing for a maximum frequency offset of f_m uses a time weighting to broaden the frequency response of the correlator (see Figure 3). The tradeoff is that the peak of the correlation detector is significantly reduced when the frequency offset is large.

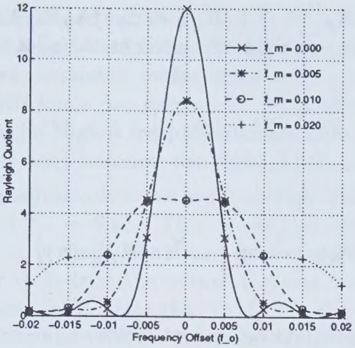


Figure 3: Ideal frequency response of the Rayleigh Quotient

5 Synchronization in Rayleigh Fading

5.1 ML Estimation

In the fading channel environment (3), the observation vector is conditionally zero mean Gaussian and the ML estimate is given by (see Appendix B)

$$\hat{J} = \underset{J\in[0,L-1]}{\text{Argmax}} \sum_{\mathbf{d}\in\Omega_d^{L-N}} \frac{1}{\left|\mathbf{R}_{\mathbf{x/d}}\right|} \exp\left\{-\left[\mathbf{T}^{(J)H}\mathbf{x}\right]^H \mathbf{R}_{\mathbf{x/d}}^{-1}\left[\mathbf{T}^{(J)H}\mathbf{x}\right]\right\} \tag{15}$$

where $\mathbf{R}_{\mathbf{x/d}}$ is the correlation matrix for the observation vector conditioned on the data if the pilots start at position 0, $\mathbf{R}_c$ is the normalized correlation matrix for the fading process, and $\mid\mathbf{A}\mid$ denotes the determinant of the matrix $\mathbf{A}$.

$$\mathbf{R}_{\mathbf{x/d}} = E_s\sigma_c^2\mathbf{\Lambda}_{\mathbf{y}}\mathbf{R}_c\mathbf{\Lambda}_{\mathbf{y}}{}^H + N_o\mathbf{I} \tag{16}$$

$$\mathbf{\Lambda}_{\mathbf{y}} = \text{Diag}\{\, y_0,\, y_1,\, \ldots,\, y_{L-1}\,\} \tag{17}$$

Unlike the AWGN case, the observations at data times cannot be neglected, i.e the optimum synchronizer uses the entire L-length observation, not just an N-length subset.

5.2 Hypothesis Testing

Once again consider hypothesis testing, and to reduce complexity ignore the observations at data times. Using the same notation as before, the relevant hypotheses are given by

$$\begin{array}{llrcl} \mathrm{H}_0: & y(n) \text{ is a data symbol} & \mathbf{z}(n) & = & \sqrt{E_s}\,\mathbf{\Lambda}_{\mathbf{d}}\mathbf{c}+\mathbf{n} \\ \mathrm{H}_1: & y(n) \text{ is } p_0 & \mathbf{z}(n) & = & \sqrt{E_s}\,\mathbf{\Lambda}_{\mathbf{p}}\mathbf{c}+\mathbf{n} \end{array}$$

where $\mathbf{\Lambda}_{\mathbf{d}}$ is a diagonal matrix of data symbols. The likelihood ratio is

$$L(n) = \frac{\left|\mathbf{R}_{\mathbf{z/p}}\right|^{-1}\exp\left\{-\mathbf{z}^H(n)\mathbf{R}_{\mathbf{z/p}}^{-1}\mathbf{z}(n)\right\}}{\frac{1}{M^N}\sum_{\mathbf{d}\in\Omega_d^N}\left|\mathbf{R}_{\mathbf{z/d}}\right|^{-1}\exp\left\{-\mathbf{z}^H(n)\mathbf{R}_{\mathbf{z/d}}^{-1}\mathbf{z}(n)\right\}} \tag{18}$$

where $\mathbf{R}_{\mathbf{z}/\mathbf{p}} = E_s\sigma_c^2 \mathbf{\Lambda}_\mathbf{p} \mathbf{R}_{c_p} \mathbf{\Lambda}_\mathbf{p}{}^H + N_o\mathbf{I}$, $\mathbf{R}_{\mathbf{z}/\mathbf{d}} = E_s\sigma_c^2 \mathbf{\Lambda}_\mathbf{d} \mathbf{R}_{c_p} \mathbf{\Lambda}_\mathbf{d}{}^H + N_o\mathbf{I}$, and $\mathbf{R}_{c_p}$ denotes the $N \times N$ normalized fading correlation matrix at the pilot times. Taking the log yields

$$-\mathbf{z}^H(n)\mathbf{R}_{\mathbf{z}/\mathbf{p}}^{-1}\mathbf{z}(n) - g(n) \underset{\mathrm{H}_0}{\overset{\mathrm{H}_1}{\gtrless}} \ln\left|\mathbf{R}_{\mathbf{z}/\mathbf{p}}\right| - N\ln M \tag{19}$$

$$g(n) = \ln\left[\sum_{\mathbf{d}\in\Omega_d^N} \left|\mathbf{R}_{\mathbf{z}/\mathbf{d}}\right|^{-1} \exp\left\{-\mathbf{z}^H(n)\mathbf{R}_{\mathbf{z}/\mathbf{d}}^{-1}\mathbf{z}(n)\right\}\right]. \tag{20}$$

As in the AWGN case, ignore data dependent terms and normalize by the received signal power (much more important in fading channel applications). This yields the following threshold test

$$r(n) = \frac{\mathbf{s}^H(n)\left(\mathbf{R}_{c_p} + \frac{N_o}{E_s\sigma_c^2}\mathbf{I}\right)^{-1}\mathbf{s}(n)}{\mathbf{s}^H(n)\mathbf{s}(n)} \underset{\mathrm{H}_1}{\overset{\mathrm{H}_0}{\gtrless}} T_h. \tag{21}$$

Care must be taken when computing this correlation at high SNR since fading processes are inherently bandlimited. It appears that the synchronizer requires knowledge of the SNR, but experimentation has shown that at high SNR performance is not sensitive to the value used in (21). Note that the same synchronizer structure is used for both an AWGN channel with a frequency offset and a slow fading channel. This occurs because slow fading can be viewed as a slowly time varying frequency offset and a slowly time varying amplitude distortion. Normalizing by the received energy removes the amplitude variations, and the quadratic structure compensates for the frequency offset.

6 Simulations

Unless stated otherwise, all simulations use a 16-QAM data set and the following 8-PSK pilot sequence with $N_p = 10$

$$\mathbf{p} = \exp(j2\pi\mathbf{k}/8) \qquad \mathbf{k} = [\,4\;3\;2\;2\;5\;7\;1\;5\;4\;0\;6\;5\,]^T. \tag{22}$$

Figures 4 and 5 show the response of the Rayleigh Quotient for an AWGN channel with a frequency offset. They also plot threshold values of $0.5 * \sum(R)/N$ and $0.9 * \sum(R)/N$. Figure 4 fixes SNR at 10 dB and shows two values of f_m, the maximum frequency offset for which the detector is designed. Notice that as f_m increases the peaks at the pilot times decrease. Figure 5 fixes $f_o = 0.005$ and $f_m = 0.01$ and shows two values of SNR. For low SNR a larger threshold will result in a very long time to acquisition, but for high SNR this threshold will yield a higher probability of false lock. These figures suggest that the threshold should be adjusted dynamically.

Figure 6 shows simulated performance versus SNR for several values of A, B, and T_h with 10,000 trials per SNR value. For each trial the actual frequency offset was generated randomly between $-f_m/2$ and $f_m/2$. In all but one case, correct acquisition was achieved in every trial, and in all cases $\mathrm{E}\{\, N_{acq} \,\}$ approached $(A+1)L$ which is the number of symbols required to perform all correlations in the first and second stages.

Note that as the threshold increased so did E$\{N_{acq}\}$. This shows that the synchronizer can be designed to achieve an arbitrarily high Pr$\{acq\}$, but it might result in long time to acquire. Figure 7 shows simulated performance versus maximum frequency offset with 10 dB SNR and 10,000 trials per frequency. Note that increasing the threshold yields a better performance for higher frequency offsets, but it takes longer to acquire. Furthermore, when Pr{False acq} becomes too high, E$\{N_{acq}\}$ is meaningless.

Figure 8 shows the quadratic detector and Rayleigh Quotient designed for a fading channel with 5dB SNR and $f_D = 0.01$. This shows that normalizing by the received energy is important. Figure 9 shows simulated performance versus SNR with 10,000 trials per SNR value. For each trial the actual Doppler frequency, f_d was generated randomly between $-f_D/2$ and $f_D/2$. Note that the fading channel requires a higher SNR to achieve good Pr{acq} than the AWGN channel. Recall that this threshold detector looks for a minimum, so lower thresholds yield a higher Pr{acq}. However, as SNR increases so do the peaks because $\mathbf{R}$ becomes singular. Therefore they might not drop below the threshold and a longer acquisition time results, i.e., a higher threshold acquires quicker, but needs a larger SNR to get good performance. This suggests the use of a dynamically adjusted threshold.

7 Conclusion

A PSAM frame synchronizer has been developed for AWGN channels with a small frequency offset and for flat Rayleigh fading channels. The synchronizer computes a Rayleigh Quotient which is tailored to the second order characteristics of the channel. Furthermore, it can be designed to achieve an arbitrarily high probability of correct acquisition, while maintaining a low computational complexity.

A Deriving the ML Estimate (AWGN)

Recall that the problem is to estimate the index J which corresponds to the start of a superframe and ML estimate is given by

$$\hat{J} = \underset{J\in[0,L-1]}{\text{Argmax}} \; f(\mathrm{x}/J) = \underset{J\in[0,L-1]}{\text{Argmax}} \int_{-\pi}^{\pi} f(\mathrm{x}/J,\theta)d\theta. \tag{23}$$

Since d_k and n_k are i.i.d. and independent of each other, the pdf can be written as

$$f(\mathrm{x}/J,\theta) = \prod_{l=0}^{N-1} f(x_{lN_p+J}/J,\theta,y_{lN_p+J} = p_l) \prod_{\substack{k=0\\k\neq lN_p}}^{L-1} \sum_{d\in\Omega_d} f(x_{k+J}/J,\theta,y_{k+J} = d)p(d) \tag{24}$$

where $p(d)$ is the pmf of d. When conditioned on y_k and θ, x_k is complex Gaussian and

$$\begin{aligned} f(\mathrm{x}/J,\theta) &= \prod_{l=0}^{N-1}\left[\frac{1}{\pi N_o}\exp\left\{-\frac{\left|x_{lN_p+J}-\sqrt{E_s}p_l e^{j\theta}\right|^2}{N_o}\right\}\right] \bullet \\ &\prod_{\substack{k=0\\k\neq lN_p}}^{L-1}\left[\frac{1}{M}\sum_{d\in\Omega_d}\frac{1}{\pi N_o}\exp\left\{-\frac{\left|x_{k+J}-\sqrt{E_s}d e^{j\theta}\right|^2}{N_o}\right\}\right]. \end{aligned} \tag{25}$$

Terms independent of J can be grouped into a constant C. Extending the product to cover all $k = 0, .., L-1$ makes it independent of J, so it can be removed. Finally, the sum and product can be interchanged if the sum is taken over all possible data vectors.

$$f(\mathbf{x}/J,\theta) \; = \; C \, \frac{\exp\left(\frac{2\sqrt{E_s}}{N_o}\,\mathcal{R}e\left\{\sum_{l=0}^{N-1} x_{lN_p+J}\, p_l^* e^{-j\theta}\right\}\right)}{\sum_{\mathbf{d}\in\Omega_d^N} \exp\left(-\frac{E_s\|d\|^2}{N_o}\right)\exp\left(\frac{2\sqrt{E_s}}{N_o}\,\mathcal{R}e\left\{\sum_{l=0}^{N-1} x_{lN_p+J}\, d^* e^{-j\theta}\right\}\right)} \qquad (26)$$

Integrating this yields the desired result (7).

B Deriving the ML Estimate (Fading)

In the fading channel model (3), a fixed observation vector can be written as

$$\mathbf{x} \; = \; \mathbf{T}^{(J)}\boldsymbol{\Lambda}_{\mathbf{y}}\mathbf{T}^{(J)H}\mathbf{c} + \mathbf{n} \qquad (27)$$

where $\boldsymbol{\Lambda}_{\mathbf{y}}$ is a diagonal matrix of pilot and data symbols with $y_0 = p_0$ and $\mathbf{T}^{(J)}$ is a rotation matrix. This is conditionally zero mean Gaussian with correlation

$$\mathbf{R}_{\mathbf{x}/J,\mathbf{d}} \; = \; \mathbf{T}^{(J)}\left[\sigma_c^2\boldsymbol{\Lambda}_{\mathbf{y}}\mathbf{R}_c\boldsymbol{\Lambda}_{\mathbf{y}}{}^H + N_o\mathbf{I}\right]\mathbf{T}^{(J)H} \; = \; \mathbf{T}^{(J)}\mathbf{R}_{\mathbf{x}/\mathbf{d}}\mathbf{T}^{(J)H} \qquad (28)$$

where $[\mathbf{R}_c]_{k,l} = J_0(2\pi k f_D)$ is the normalized correlation for the fading process and $\mathbf{R}_{\mathbf{x}/\mathbf{d}}$ is the correlation matrix for $J = 0$. Note that $\left|\mathbf{R}_{\mathbf{x}/J,\mathbf{d}}\right| = \left|\mathbf{R}_{\mathbf{x}/\mathbf{d}}\right|$, so the pdf of $\mathbf{x}$ is

$$f(\mathbf{x}/J) \; = \; \frac{1}{M^{L-N}} \sum_{\mathbf{d}\in\Omega_d^{L-N}} \frac{1}{\pi^L\left|\mathbf{R}_{\mathbf{x}/J,\mathbf{d}}\right|}\exp\left\{-\mathbf{x}^H\mathbf{R}_{\mathbf{x}/J,\mathbf{d}}^{-1}\mathbf{x}\right\} \qquad (29)$$

$$= \; \frac{1}{M^{(L-N)}\pi^L} \sum_{\mathbf{d}\in\Omega_d^{L-N}} \frac{1}{\left|\mathbf{R}_{\mathbf{x}/\mathbf{d}}\right|}\exp\left\{-\left[\mathbf{T}^{(J)H}\mathbf{x}\right]^H\mathbf{R}_{\mathbf{x}/\mathbf{d}}^{-1}\left[\mathbf{T}^{(J)H}\mathbf{x}\right]\right\}. \qquad (30)$$

References

[1] James K. Cavers. An analysis of pilot symbol assisted modulation for Rayleigh fading channels. *IEEE Trans. Vehicular Tech.*, 40(4):686–693, Nov. 1991.

[2] James L. Massey. Optimum frame synchronization. *IEEE Trans. Commun.*, COM-20(2):115–119, April 1972.

[3] Gee L. Lui and Harry H. Tan. Frame synchronization for Gaussian channels. *IEEE Trans. Commun.*, COM-35(8):818–829, Aug. 1987.

[4] Patrick Robertson. Maximum Likelihood frame synchronization for flat fading channels. *ICC*, ICC-92(5):1426–1430, 1992.

[5] Andreas Polydoros and Charles L. Weber. A unified approach to serial search spread-spectrum code acquisition – Parts I & II. *IEEE Trans. Commun.*, COM-32(5):542–560, May 1984.

[6] W. C. Jakes. *Microwave Mobile Communications.* Wiley, New York, 1974.

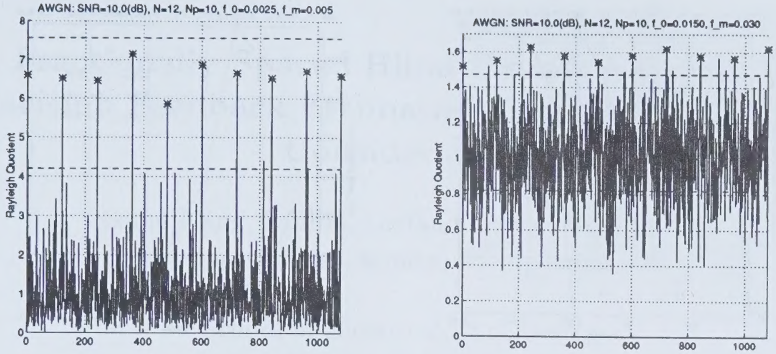


Figure 4: Rayleigh Quotient for AWGN with $f_m = 0.005$, $f_m = 0.03$ SNR = 10dB

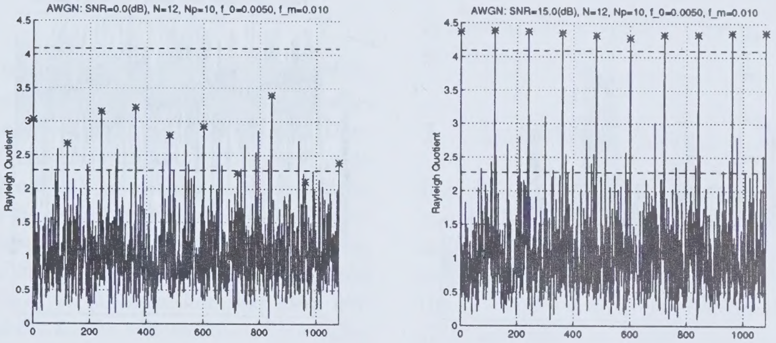


Figure 5: Rayleigh Quotient for AWGN with SNR = 0dB, SNR = 12dB when $f_m = 0.01$

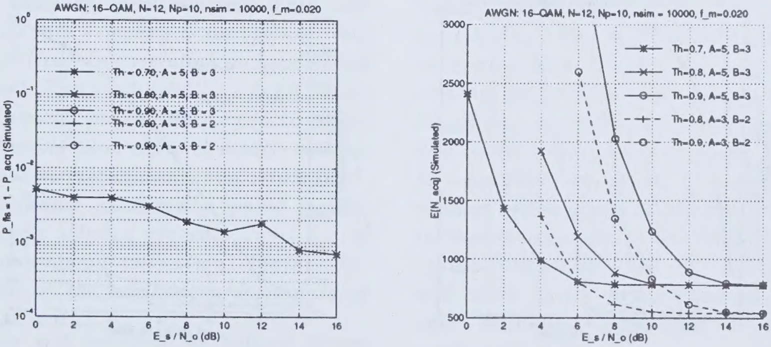


Figure 6: Pr{False acq} and E{ N_{acq} } vs. SNR for AWGN with $f_m = 0.020$.

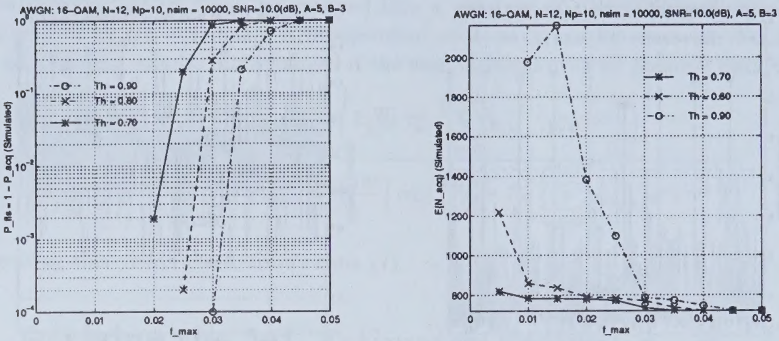


Figure 7: Pr{False acq} and E{ N_{acq} } vs. f_m for AWGN with SNR = 10dB

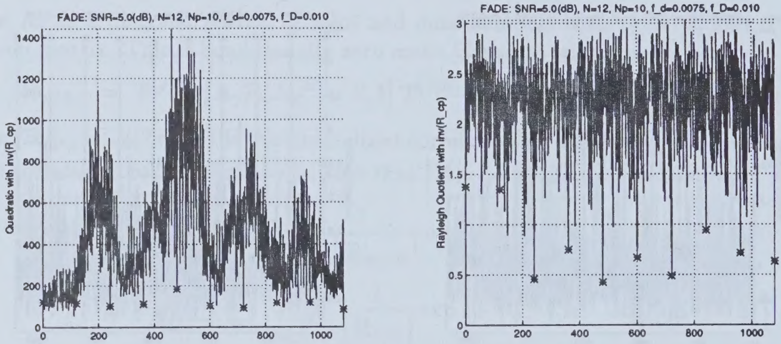


Figure 8: Quadratic and Rayleigh Quotient for a fading channel with $f_D = 0.01$

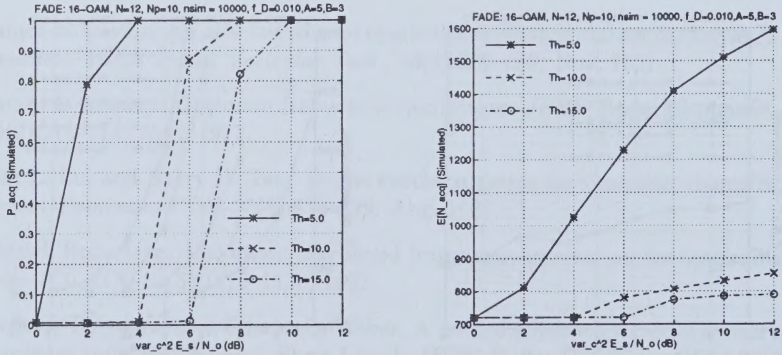


Figure 9: Pr{False acq} and E{ N_{acq} } vs. SNR for a fading channel with $f_D = 0.01$

A Fractionally Spaced Blind Bayesian Conditional Decision Feedback Estimator with Symbol Timing Consideration*

Insung Kang, Saul B. Gelfand and Michael P. Fitz
{insung, gelfand, mpfitz}@ecn.purdue.edu

School of Electrical Engineering
Purdue University
W. Lafayette, IN 47907-1285

Abstract

In this work, we consider the effect of the symbol timing error on the performance of the blind Bayesian Conditional Decision Feedback Estimator [2]. First optimal front-end processing is discussed because the effect of the symbol timing error depends on the front-end processing. When the effective transmission data pulse is defined as the ideal low pass filter output to the known transmission data pulse, it is shown that the fractionally spaced whitened matched filter matched to the effective transmission data pulse is optimal and is realized by an ideal low pass filter. Thus an ideal low pass filter with Nyquist rate sampling is an optimal front end for blind deconvolution on bandlimited channel. To investigate the symbol timing effect on the performance, we use different discrete time channel models suitable for noncoherent and coherent estimation[1]*. Unlike linear fractionally spaced equalizers, cycle slipping prevents direct use of the noncoherent estimation. Further discussion of the advantages/disadvantages of each model is presented along with experimental results.*

1 Introduction

It is well known that the optimum front-end structure for maximum likelihood sequence estimation (MLSE) of a linearly modulated uncoded sequence in the presense of intersymbol interference (ISI) and additive white Gaussian noise (AWGN) consists of a matched filter (MF) matched to the combined transmission data pulse and channel response followed by a whitening filter (WF) [1], usually called the whitened matched filter (WMF)[2]. The WMF output, sampled at symbol rate, 1) provides a sufficient statistic with white noise components. 2) transforms a continuous time (possibly) non-causal non-minimum phase channel to a discrete time causal minimum phase channel[3]. The phase of the resulting discrete time channel is not important to optimal estimators but has a considerable effect on the performance of suboptimal estimators using decision feedback

---

*This work supported by National Science Foundation under contract NCR-9406073.

[1] Noncoherent estimation implies no explicit estimation of timing offset; coherent estimation implies explicit estimation of timing offset.

[2] The whitening filter can be implemented in discrete time under certain (invertibility) conditions.

[3] causal ≡ onesided.

[2],[3],[4],[5].

When the channel is *a priori* unknown it is not possible to use a matched filter based on the channel response. It is known that the symbol-rate samples of the output of a matched filter matched to the transmission pulse provide a sufficient statistic if the transmission pulse is a Nyquist pulse at the symbol rate and the received signal dispersed by the channel can be expressed as a weighted sum of the delayed transmission pulses [7, ch4]. This is a very strong restriction on the channel. Recently Chugg [6] showed that a fractionally spaced whitened matched filter (FS-WMF) matched to the known transmission data pulse (possibly non-Nyquist) provides a sufficient statistic for MLSE on tapped delay line channel model when channel is unknown. The required sampling rate is determined by the channel bandwidth for bandlimited channels and should be an integer (≥ 2) multiple of the symbol rate for the resulting discrete-time channel to be time-invariant.

For bandlimited channels, the transmission data pulse which Chugg used in his derivation is not adequate one. Instead, we found that the effective transmission data pulse should be used. The effective transmission data pulse is defined as an ideal low pass filter (LPF) output to the transmission pulse, where the bandwidth of the ideal LPF is $1/2T_s$. It is shown that the FS-WMF matched to the effective transmission data pulse is an optimal front end and is realized by an ideal LPF, magnitude of which is scaled by the square root of the sampling interval. Thus we use the FS-WMF matched to the effective transmission data pulse (equivalently LPF) as a front-end filter, and present discrete-time channel models which account for fractional sampling and unknown symbol timing. The blind Bayesian conditional decision feedback estimator (BCDFE) can be applied to these models with minor modifications.

The remainder of this paper is organized as follows. In section 2, the optimal front-end processing is discussed for blind deconvolution on bandlimited channels. In section 3, the symbol timing error, which causes the WF to mismatch and produce additional ISI, is considered. Two different vector channel models are presented which are suitable for noncoherent and coherent estimation. In section 4, simulation results are presented for each model. Finally, conclusions are given in section 5. The blind BCDFE algorithm is not listed for brevity (See [2] for details).

2 Front-end Processing

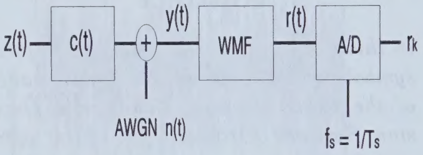


Figure 1: A digital communication system model.

Consider a digital communication system model shown in Fig. 1, where $z(t)$ is the complex baseband transmitted signal and $c(t)$ is continuous waveform bandlimited channel. We will show that the FS-WMF matched to the effective transmission data pulse is an optimal front end. One subtle point which we have to be aware of is that the bandwidth of the transmission data pulse must be large enough to match the sampling rate: i.e.,

$$|U(f)| > 0 \quad for\ |f| \leq \frac{1}{2T_s} \qquad (1)$$

where $U(f)$ is the Fourier transform of the transmission data pulse. This is a sufficient condition for the existence of the WF.

Here we show that the MF should match the effective transmission data pulse when the channel is bandlimited. The two-sided bandwidth of the channel is denoted by B. The constraints on the sampling rate $1/T_s$ are:

$$1/T_s \geq B$$

and

$$N_r = T/T_s$$

where N_r is an integer ($N_r \geq 2$) and $1/T$ is the symbol rate.

We consider only a linearly modulated uncoded sequence transmission. Thus the complex baseband transmitted signal, $z(t)$, is given by:

$$z(t) = \sum_{n=0}^{N} d_n\, u(t - nT) \qquad (2)$$

where N is the number of the transmitted symbol, d_n are complex transmitted symbols, and $u(t)$ is a known transmission pulse. The complex baseband received signal, $y(t)$, is given by:

$$y(t) = \sum_{n=0}^{N} d_n\, h(t - nT) + n(t) \qquad (3)$$

where the overall channel impulse response is the convolution $h(t) = c(t) * u(t)$ and $n(t)$ is complex AWGN with unilateral spectral density of N_0.

By Nyquist sampling theorem, $c(t)$ can be expressed as:

$$c(t) = \sum_{l=-\infty}^{\infty} c_l\, \frac{1}{T_s}\, sinc(\frac{t - lT_s}{T_s}) \qquad (4)$$

where $c_l = T_s\, c(lT_s)$. For all practical purposes, $c(t)$ can be assumed to be time-limited also and the amount of information loss can be made negligible. Hence

$$c(t) = \sum_{l=0}^{L_c-1} c_l\, \frac{1}{T_s}\, sinc(\frac{t - lT_s}{T_s}) \qquad (5)$$

where L_c is the (sufficiently long) channel length. Thus

$$\begin{aligned} h(t) &= c(t) * u(t) \\ &= \sum_{l=0}^{L_c-1} c_l\, g(t - lT_s) \qquad (6) \end{aligned}$$

where

$$\begin{aligned} g(t) = &\ \tfrac{1}{T_s} \textstyle\int_{-\infty}^{\infty} sinc(\tfrac{t-s}{T_s})\, u(s)\, ds \\ = &\ \textstyle\sum_{m=-\infty}^{\infty} g_m \tfrac{1}{\sqrt{T_s}} sinc(\tfrac{t-mT_s}{T_s}) \qquad (7) \end{aligned}$$

and $g_m = \sqrt{T_s}\, g(mT_s)$. Substituting Eq. (6) into Eq. (3), we have

$$y(t) = \sum_{n=0}^{N} d_n \sum_{l=0}^{L_c-1} c_l g(t - N_r n T_s - lT_s) + n(t). \qquad (8)$$

Since $g(t - kT_s)$ are a finite set of basis functions which constitute the received signal $y(t)$, the sampled outputs of a filter $g^*(-t)$ matched to $g(t)$ form a sufficient statistic [1].[4] That is, the MF matches $g(t)$ which is an ideal LPF output to the transmission pulse $u(t)$. $g(t)$ is the effective transmission pulse.

Now we show that an ideal LPF with the bandwidth of $1/2T_s$ is a class of the FS-WMF matched to the effective transmission pulse. The existence of the WF is guaranteed by Eq. (1). We denote inner-product by

$$< a(t), b(t) > = \int_{-\infty}^{\infty} a(t) b^*(t)\, dt.$$

Let

$$w_m(t) = 1/\sqrt{T_s} sinc((t - mT_s)/T_s).$$

Note that $w_m(t)$ are a set of orthonormal basis functions. Then the pulse autocorrelation coefficients of $g(t)$ is given by:

$$\begin{aligned} R_k &= < g(t), g(t - kT_s) > \\ &= < \sum_n g_n w_n(t), \sum_m g_m w_{k+m}(t) > \\ &= \sum_m g_{k+m} g_m^*. \qquad (9) \end{aligned}$$

Thus the pulse autocorrelation function, $R_{gg}(z)$, is given by:

$$\begin{aligned} R_{gg}(z) &= \sum_k R_k z^{-k} \\ &= G(z) G^*(z^{-*}) \qquad (10) \end{aligned}$$

---

[4] The sampling rate is $1/T_s$ because the basis functions are T_s-spaced.

where

$$G(z) = \sum_m g_m z^{-m}.$$

$G(z)$ is the Z-transform[5] of g_m. We choose $1/G^*(z^{-*})$ for the WF (we assume that arbitrary length of delay is available to make the WF realizable.). Then the WMF output sampled at the rate of $1/T_s$ is $G(z)$ which is equivalent to g_m. On the other hand, the WMF noise output n_k sampled at the rate of $1/T_s$ is also complex AWGN with the variance of N_0 when the WMF noise input $n(t)$ is complex AWGN with unilateral spectral density of N_0.

An ideal LPF with the magnitude being scaled by $\sqrt{T_s}$ in the passband and having the bandwidth of $1/2T_s$ provides the same sampled signal to the input $g(t)$ and sampled noise sequence having the same variance. Thus an ideal LPF is equivalent to the FS-WMF matched to the effective transmission pulse with a choice of the WF $1/G^*(z^{-*})$. Therefore an ideal LPF followed by Nyquist-rate sampler is an information lossless front-end processing. The Nyquist-rate sampled outputs of an ideal LPF can be used to construct the symbol-rate sufficient statistic adaptively using discrete time matched filter [9, ch10].

3 Vector Channel Models

The received signal after the FS-WMF matched to the effective transmission pulse or equivalently an ideal LPF is given by:

$$r(t) = \sum_{n=0}^{N} d_n \sum_{l=0}^{L_c-1} c_l g(t - N_r n T_s - l T_s) + n(t). \tag{11}$$

The discrete time equivalent signal model with symbol timing offset is given by sampling $r(t)$ as follows:

$$\begin{aligned} r_k = & \sum_n d_n \sum_{l=0}^{L_c-1} c_l g((k - N_r n - l)T_s + \tau) \\ & + n_k \qquad (12) \\ = & \sum_{l=0}^{L_c-1} c_l \sum_n d_n g((k - N_r n - l)T_s + \tau) \\ & + n_k \qquad (13) \end{aligned}$$

where $N_r = T/T_s$ is an integer number representing the fractional sample spacing, and T is the symbol interval, τ represents the symbol timing offset, and n_k is complex AWGN.

3.1 Vector Channel Model for Noncoherent Estimation

Eq. (12) can be expressed in vector notation as follows:

$$\mathbf{r}_k = \mathbf{D}(k)\mathbf{h} + \mathbf{n}_k \tag{14}$$

where

$$\mathbf{r}_k = [r_{N_r k} \quad r_{N_r k+1} \quad \cdots \quad r_{N_r k+N_r-1}]^T$$

$$\mathbf{D}(k) = \begin{bmatrix} d_k\, 0 \cdot\cdot\, d_{k-1} & 0 & \cdots d_{k-L} & 0 & \cdots & 0 \\ 0\, d_k\, 0 & \cdots d_{k-1} & 0 & \cdots d_{k-L}\, 0 & \cdots & \\ \vdots & & \ddots & & & \vdots \\ 0 \cdots d_k & 0 & \cdots d_{k-1} & 0 & \cdots & d_{k-L} \end{bmatrix}$$

$$\mathbf{h} = [h_0 \quad h_1 \quad \cdots \quad h_{N_r * L-1}]^T$$

$$\mathbf{n}_k = [n_{N_r k} \quad n_{N_r k+1} \quad \cdots \quad n_{N_r k+N_r-1}]^T.$$

Here the number of the fractional channel taps, $N_r * L$, is assumed to be known, usually $L = L_c + L_s - 1$ where L_s is the number of significant samples in $g(t)$.

The advantages of this model are 1) the observation matrix $D(k)$ does not depend on the symbol timing offset, therefore is completely known in the optimal blind BCDFE, 2) the subchannels are independent, therefore the computational complexity can be reduced. The disadvantage is that it could be difficult to detect a cycle slip[6] since the symbol timing offset is not estimated. One possible way to detect the cycle slip is to monitor the ratio of the moving averaged channel energy between the channel taps corresponding to d_k and the channel taps corresponding to d_{k-1}. If that ratio is under a certain threshold, a cycle slip is assumed to be detected. Then estimation algorithm proceeds into recovering action. One good alternative is explained in the next section, i.e. coherent estimation.

---

[5] Here the Z-transform is T_s-spaced.

[6] Here, cycle slip is used to mean a drift in channel estimation.

3.2 Vector Channel Model for Coherent Estimation

The vector channel model for coherent estimation is slightly different from that for noncoherent estimation. This model was used in [8]. However, his derivation lacks the consideration of the optimality of the front-end processing. As a result, it could lose too much information, especially when the excess bandwidth is large enough. By expressing Eq. (13) in vector notation, we get

$$\mathbf{r}_k = \mathbf{H}(k;\tau)\mathbf{c} + \mathbf{n}_k \qquad (15)$$

where

$$\mathbf{r}_k = [r_{N_r k} \quad r_{N_r k+1} \quad \cdots \quad r_{N_r k+N_r-1}]^T$$

$$\mathbf{H}(k;\tau) = \begin{bmatrix} H_{11}(k;\tau) & \cdots & H_{1L_c}(k;\tau) \\ \vdots & \ddots & \vdots \\ H_{N_r 1}(k;\tau) & \cdots & H_{N_r L_c}(k;\tau) \end{bmatrix}$$

$$\begin{aligned} H_{jl}(k;\tau) &= \sum_{n=0}^{N} d_n \cdot \\ & \qquad g(((k-n)N_r + j - l)T_s + \tau) \\ & \qquad \forall \quad j = 1,\cdots,N_r \ , \ l = 1,\cdots,L_c \end{aligned}$$

$$\mathbf{c} = [c_0 \quad c_1 \quad \cdots \quad c_{L_c-1}]^T$$

$$\mathbf{n}_k = [n_{N_r k} \quad n_{N_r k+1} \quad \cdots \quad n_{N_r k+N_r-1}]^T .$$

The advantages/disadvantages of this model are just opposite to those of the previous model. Since the observation matrix is nonlinearly dependent on the symbol timing offset, linearization by Taylor series expansion is necessary. This linearization is a major factor to limit the application of this model to a certain type of transmission waveforms. For example, there exist a lot of ambiguities in the derivative of the pseudo noise (PN) waveform in direct sequence spread spectrum system. Once tracking falls into the false stable point, it is difficult to come back to the true stable point.

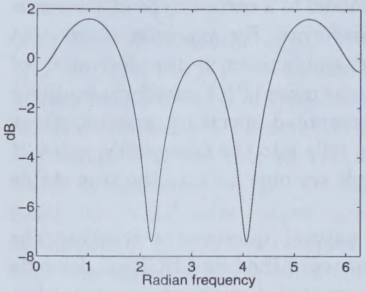
One natural question is whether the performance of the blind BCDFE depends on the symbol timing no matter what model is used. It is considered that the performance is a function of the pulse energy and the phase of the effective discrete time channel. Since the pulse energy and the channel phase is generally a function of the symbol timing, the performance of the blind BCDFE depends on the symbol timing especially when suboptimal algorithm is used to reduce computational complexity. Though the performance dependence of the BCDFE on the symbol timing can be minimized with careful design of the transmission pulse, the pulse design problem is not included here.

4 Performance

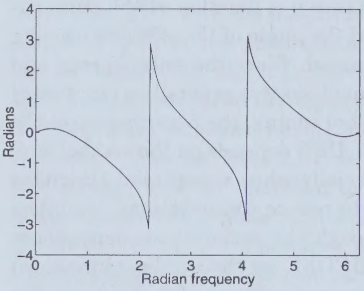
A non-minimum phase channel is considered, the impulse response of which is approximated by

$$c(t) = \sum_{l=0}^{3} c_l \frac{1}{T_s} sinc(\frac{t - lT_s}{T_s})$$

where c_l = [.67 .47 .26 -.52]. The frequency response of the channel is plotted in Fig. 2. Bandwidth efficient signaling is assumed, i.e. $N_r = 2$. Differentially encoded Gray coded QPSK is used for all simulations. The magnitude of the transmission pulse is normalized to have unit pulse correlation at time 0. The sample transmission pulse used in the simulation is shown in Fig. 3. Ideal low pass filter with the bandwidth of $1/2T_s$ is used as the front-end filter. We use the letters J



(a) Magnitude response



(b) Phase response

Figure 2: Channel characteristic.

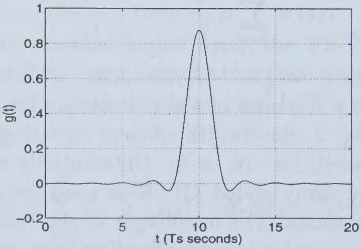


Figure 3: Sample transmission pulse.

and K to denote the chip length [7] and the decision delay, respectively.

In Fig. 4, the learning characteristic of noncoherent blind deconvolution with the symbol timing offsets of 0 and $-T_s/2$ is shown. Note that the bit error probability (BEP) performance for the timing offset of $-T_s/2$ is bounded by about 10^{-2}. This is due to the cycle slip phenomenon. In Fig. 5 and Fig. 6, typical characteristics of the cycle slip of the BCDFE are shown. To accommodate the cycle slip, the number of the channel tap estimates used is N_r more than the one actually required. For the channel given, the $(J = 2)$ BCDFE corresponds to the MAP estimator and the $(J = 0)$ BCDFE corresponds to the decision feedback equalizer (DFE).

In Fig. 5 (a) and Fig. 6 (a) the BEP curves are shown where solid lines show the performance degraded by the cycle slip and dashed lines show the performance with the delayed comparison by assuming that the cycle slip is detected but no recovering action is taken. When the cycle slip occurs, the effective chip length of the BCDFE reduces by 1 because the channel tap estimates corresponding to the incoming symbol, d_k, shrink to zeros. Thus performance degrades even with the delayed comparison. As is shown in Fig. 5 (b)-(c) and Fig. 6 (b)-(c) where solid lines are the channel tap estimates and dotted lines are the real channel taps, the channel tap estimates are shifted in one symbol, thus the channel tap estimates corresponding to the incoming symbol d_k shrink to zero. Note that the cycle clip is recovered by itself when the chip length J is equal to 1, thus the effective chip length after the cycle slip reduces to 0. This self-recovery from the cycle slip can be condsidered another occurrence of the cycle slip. Since the reduction of the effective chip length of the BCDFE corresponds to lowering SNR, the cycle slip is more likely to occur after one cycle slip.

---

[7] Unlike in the spread spectrum communications, the "chip" length is used to mean the number of the look-ahead symbols in the BCDFE.

In Fig. 7 and Fig. 8, the simulation result are given to show that coherent estimation is a good solution to the cycle slip problem inherent in noncoherent estimation. One thing we would like to see is whether the performance of the J-chip BCDFE depends on the symbol timing. Though discrete time channel for the exact timing has better channel phase than the one for the timing of $T_s/2$, simulation results show that slightly better performance is achieved for the timing of $T_s/2$. This is due to the fact that the pulse energy for the timing of $T_s/2$ is bigger than that of the exact timing. This effect is prominent in the suboptimal BCDFE.

5 Conclusion

We have shown that the FS-WMF matched to the effective transmission pulse is equivalent to an ideal LPF with phase ambiguity. Two different vector channel models are presented and tested for noncoherent estimation and coherent estimation. The inherent problem for noncohent estimation, the cycle slip, is discussed through computer simulation. Coherent estimation is shown to be a good candidate for avoiding the performance degradation due to the cycle slip. The performance dependence of the BCDFE on the symbol timing is shown to be prominent in the suboptimal BCDFE through computer simulation.

References

[1] G. D. Forney, “Maximum-likelihood sequence estimation of digital sequences in the presense of intersymbol interference,” *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 363-378, May 1972.

[2] G. K. Lee, S. B. Gelfand and M. P. Fitz, “Bayesian decision feedback techniques for deconvolution,” *IEEE J. Select. Areas Commun.*, vol. SAC-13, pp. 155-166, Jan. 1995.

[3] M. V. Eyuboğlu and S. U. H. Qureshi, “ Reduced-state sequence estimation with set partitioning and decision feedback,” *IEEE Trans. Commun.*, vol. COM-36, pp. 13-20, Jan. 1988.

[4] A. Duel-Hallen and C. Heegard, “Delayed decision-feedback sequence estimation,” *IEEE Trans. Commun.*, vol. COM-37, pp. 428-436, May. 1989.

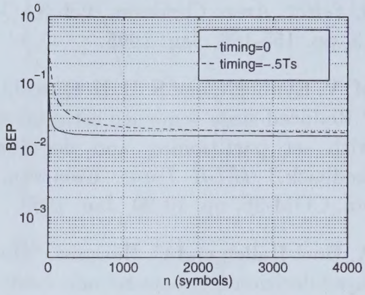
[5] K. Wesołowski, “An efficient DFE & ML suboptimum receiver for data transmission over dispersive channels using two-dimensional signal constellations,” *IEEE Trans. Commun.*, vol. COM-35, pp. 336-339, Mar. 1987.

[6] K. M. Chugg and A. Polydoros, ”Front-end processing for joint maximum likelihood channel and sequence estimation,” In *IEEE GLOBECOM commun. theory mini-conference*, pp. 51-55, Nov. 1994.

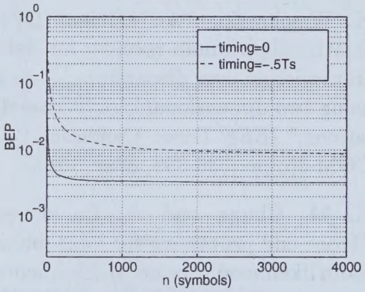
[7] R. E. Blahut, *Digital Transmission of Information*, Addison-Wesley, Reading, Mass., 1990.

[8] R. A. Iltis, “A Bayesian maximum-likelihood sequence estimation algorithm for *a priori* unknown channels and symbol timing,” *IEEE J. Select. Areas Commun.*, vol. SAC-10, pp. 579-588, Apr. 1992.

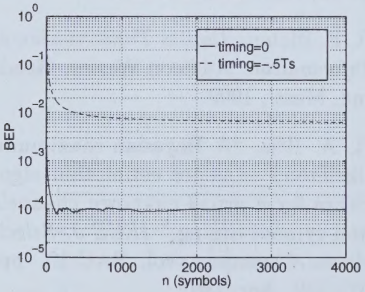
[9] E. A. Lee and D. G. Messerschmitt, *Digital Communication*, Kluwer Academic Publishers, Norwell, M.A., 1994.



(a) Bit error probability for $SNR = 5dB$

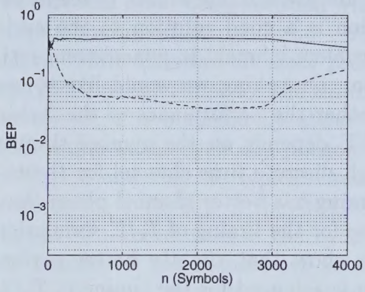


(b) Bit error probability for $SNR = 7dB$

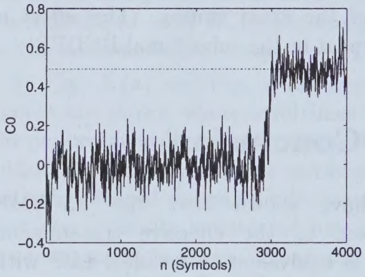


(c) Bit error probability for $SNR = 10dB$

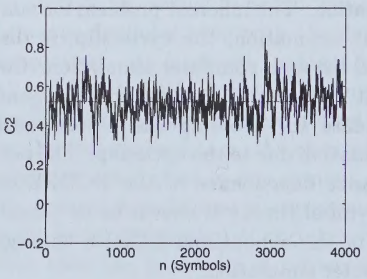
Figure 4: Learning characteristic for noncoherent estimation of the symbol timing where $J = 1$, $K = 12$, and 1000 runs of 4000 symbols.



(a) Bit error probability

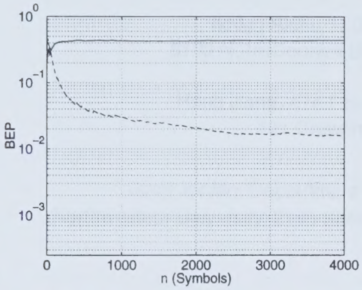


(b) The 1st channel tap estimate of subchannel 1

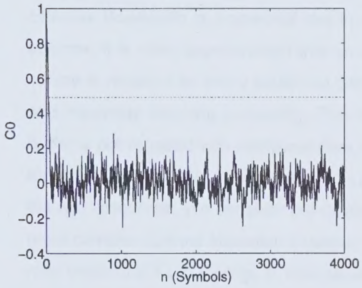


(c) The 2nd channel tap estimate of subchannel 1

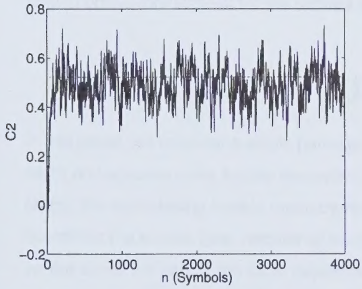
Figure 5: Cycle slipping characteristic for noncoherent estimation of the symbol timing where $J = 1$, $K = 12$, and $SNR = 5dB$.



(a) Bit error probability

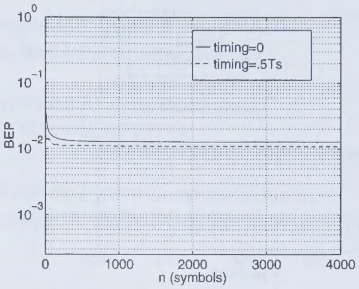


(b) The 1st channel tap estimate of subchannel 1

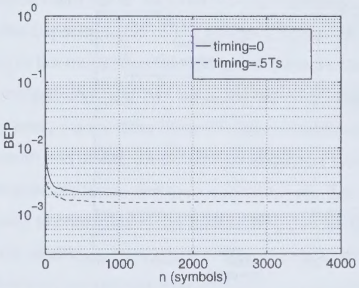


(c) The 2nd channel tap estimate of subchannel 1

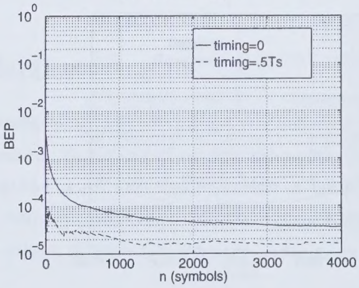
Figure 6: Cycle slipping characteristic for noncoherent estimation of the symbol timing where $J = 2$, $K = 12$, and $SNR = 5dB$.



(a) Bit error probability when $SNR = 5dB$

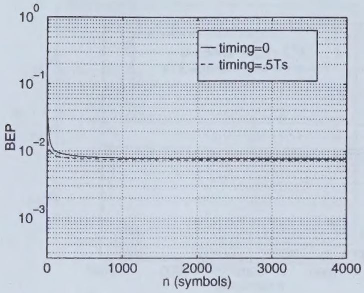


(b) Bit error probability when $SNR = 7dB$

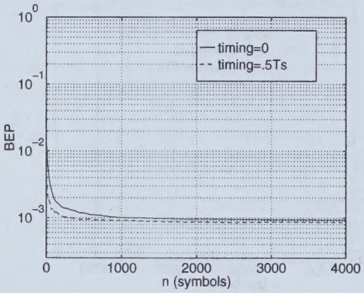


(c) Bit error probability when $SNR = 10dB$

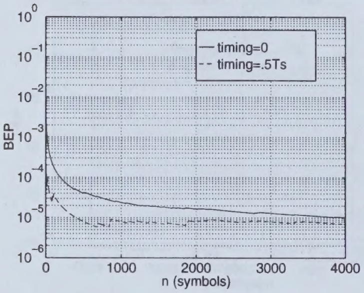
Figure 7: Simulation results for coherent estimation where $J = 1$, $K = 12$, and 1000 runs of 4000 symbols.



(a) Bit error probability when $SNR = 5dB$



(b) Bit error probability when $SNR = 7dB$



(c) Bit error probability when $SNR = 10dB$

Figure 8: Simulation results for coherent estimation where $J = 2$, $K = 12$, and 1000 runs of 4000 symbols.

Temporal Multiplexing Gains

Ikhlaq Sidhu, U.S.Robotics, Corporate Systems Division,
8100 N. McCormick Blvd, Skokie, IL 60076-2999, isidhu@usr.com

Hong Jiang, AT&T Bell Labs, Performance Analysis Department,
101 Crawfords Corner Rd, Holmdel, NJ 07733

Scott Jordan, Deptartment of Electrical Engineering and Computer Science,
Northwestern University, 2145 Sheridan Rd., Evanston, IL 60208-3118, scott@eecs.nwu.edu

Abstract

Effective Bandwidth is a powerful tool in modeling bandwidth and buffer requirements for multiplexed data sources. It is often approximated that an additional service capacity equal to the effective bandwidth of the source is required for every additional data source added to the multiplexed stream given a fixed buffer size and maximum blocking probability. This implies that a multiplexing gain can be achieved since additional buffer is not required with additional data sources. In this work, we consider other choices of bandwidth and buffer for a multiplexed case. By setting a price for bandwidth and buffer, we determine an optimal choice for the two resources. We consider the savings between simpler resource allocation schemes with that of the more complex optimal allocation scheme. We are interested to characterize the multiplexing gain so that we may determine if the savings in multiplexing outweighs the additional complexity of choosing optimal buffer and bandwidth at the multiplexor. In this regard, we find that the cost savings of optimal allocation over certain other allocation schemes is increasing and convex. By considering the case of On/Off Markov Fluids, we are able to obtain expressions for the optimal resource allocation path and resulting cost savings.

I. Introduction

In this paper, we consider a single homogenous service, multiplexing node with resources of buffer and bandwidth and assume costs for the respective resources. Based on the concept of effective bandwidth [1] [2] [3] [4] [5], the multiplexing node's resource requirements may be characterized by a bandwidth vs. buffer curve specific to the source type, number of sources, and QoS. Although, any combination of buffer and bandwidth on the curve will satisfy the QoS requirement, there is a unique, optimal operating point on the curve which minimizes the combined buffer and bandwidth costs.

We characterize the savings of optimal allocation over that using simpler allocation schemes parameterized by the number of sources being multiplexed. The three alternative schemes are Fixed Buffer, Incremented Buffer, and No Sharing. With Fixed Buffer (FB), the bandwidth resource increases by the 'effective bandwidth' with every source being multiplexed. With Incremented Buffer (IB), the buffer increases in fixed increments with the number of sources being multiplexed and the bandwidth is increased as necessary to satisfy the loss probability requirement. Finally, the No Sharing (NS) scheme considers the case when buffer and bandwidth are not shared between sources. We are interested to know if the cost benefit of optimal allocation outweighs the additional complexity. Additionally, we are interested to characterize the optimal allocation and cost savings with varying numbers of sources.

In section II, we define the model for multiplexing of homogenous sources using the schemes of optimal allocation, Fixed buffer allocation, Incremented Buffer allocation, and No Sharing allocation. This model relies on previous approximations that effective bandwidths are additive. More recent work has suggested improvements to that approximation. It is generally accepted that the additive nature of effective bandwidth is due to temporal gains of multiplexing while gains of capacity sharing are reflected in an effective bandwidth per source which decreases with the number of sources being multiplexed [6] [7]. In section III, we examine results for a wide variety of sources whose effective bandwidths are convex decreasing functions of buffer length. We give conditions for optimal allocation and proceed to show that the cost savings of optimal allocation over other allocation schemes is increasing and convex in the number of sources. In section IV, we consider Markov fluid On/Off sources and examine the multiplexing model and optimal resource allocation path as a function of number of sources.

II. Resource Allocation Model for Homogenous Multiplexing

In this section, we define a model to evaluate combinations of buffer and bandwidth at a multiplexor for differing numbers of homogenous sources given a constraint on blocking probability. Our model is based on the well-accepted concepts of effective bandwidth. By using a fixed cost for additional unit bandwidth and additional unit buffer we define an optimum resource allocation combination which varies with the number of sources being multiplexed. Further, we will compare the cost of the optimum allocation point with the cost of other allocation points obtained using simpler allocation schemes. We define the differences in the costs between these alternative schemes as cost savings which may be achieved through multiplexing.

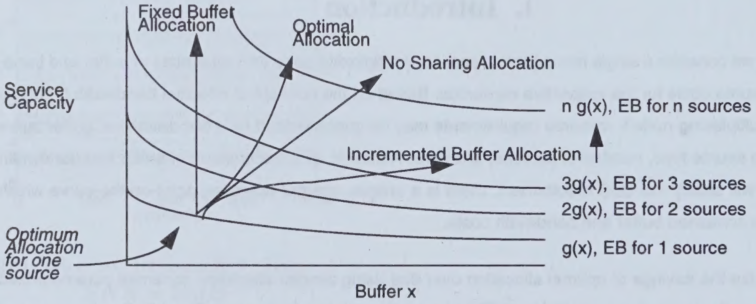


Figure 1 Resource allocation model

Consider a model to evaluate alternative buffer and bandwidth combinations parameterized by the number of sources being multiplexed (Figure 1). The curve labeled g(x) illustrates the effective bandwidth for one source as a function of buffer length x given a predefined loss constraint. We assume g(x) ranges between the peak rate and mean rate and is convex and decreasing in x [Elwa93]. Approximating the effective bandwidth of *n* sources into a buffer of length *x* as *n* times the effective bandwidth a single source into a buffer of length *x*, the corresponding curve for n sources is *n g(x)*.

To evaluate which allocation points may be more desirable, we define C_B and C_C to be the costs per unit buffer and bandwidth respectively and *m* to be the ratio C_B/C_C. We equate optimal resource allocation for *n* multiplexed sources with the least cost point on the n source effective bandwidth curve *n g(x)*. We are interested to characterize the path of optimal allocation points parameterized by *n*. The optimal cost $C^*(n)$ (EQ 3) can be obtained by the dot product of the cost vector (C_B,C_C) and the allocation point given in (EQ 2). The buffer value $x^*(n)$ which minimizes the system cost is also varying with the number of sources *n*. Define the optimal allocation point $P^*(n)$ In Figure 1, we illustrate a sample optimal allocation path.

$$x^*(n) = arg_x min \begin{bmatrix} C_B \\ C_C \end{bmatrix} \bullet \begin{bmatrix} x & ng(x) \end{bmatrix} \qquad \text{(EQ 1)}$$

$$P^*(n) = [x^*(n), ng(x^*(n))] \qquad \text{(EQ 2)}$$

$$C^*(n) = \begin{bmatrix} C_B \\ C_C \end{bmatrix} \bullet [x^*(n), ng(x^*(n))] \qquad \text{(EQ 3)}$$

Consider an allocation scheme in which buffer is fixed at $x^*(1)$ and bandwidth is set to $n\ g(x^*(1))$. In this scheme, the allocation point for one source is optimal and an additional service capacity equal to the effective bandwidth is required for every additional source. We define allocation points for this scheme of *Fixed Buffer* $P_{FB}(n)$. The path of $P_{FB}(n)$ is a vertical line (Figure 1). The cost path of $P_{FB}(n)$ is given by $C_{FB}(n)$

$$P_{FB}(n) = [x^*(1), ng(x^*(1))] \qquad \text{(EQ 4)}$$

$$C_{FB}(n) = C^*(1) + (n-1)g(x^*(1))C_C \qquad \text{(EQ 5)}$$

Alternatively, we might increment the buffer resources by x*(1) with every source and choose the corresponding capacity $n\ g(n\ x^*(1))$ which takes advantage of multiplexing gains. This scheme of Incremented Buffer is given by (EQ 6) and (EQ 7) below.

$$P_{IB}(n) = [nx^*(1), ng(nx^*(1))] \qquad \text{(EQ 6)}$$

$$C_{IB}(n) = \begin{bmatrix} C_B \\ C_C \end{bmatrix} \bullet [nx^*(1), ng(nx^*(1))] \qquad \text{(EQ 7)}$$

Finally, consider the allocation scheme consistent with no sharing of resources. The allocation points are then given by $P_{NS}(n) = (nx^*(1),n\ g(x^*(1))$ and the path constitutes a line with slope $g(x^*(1))/x^*(1)$ (Figure 1). The cost path $C_{NS}(n)$ increases linearly with *n*.

$$P_{NS}(n) = [nx^*(1), ng(x^*(1))] \tag{EQ 8}$$

$$C_{NS}(n) = nC^*(1) = nC_{FB}(1) \tag{EQ 9}$$

Since $C^*(n)$ is defined by the minimum cost allocation, then fixed buffer scheme cost $C_{FB}(n)$, the incremented buffer cost scheme $C_{IB}(n)$, and the no sharing cost $C_{NS}(n)$ must each be greater in value. We define a cost saving $\Delta C^*_{FB}(n) = C_{FB}(n)-C^*(n)$ relating optimal allocation over fixed buffer allocation, a cost savings $\Delta C^*_{IB}(n) = C_{IB}(n)-C^*(n)$ of optimal allocation over the incremented buffer and a cost savings $\Delta C^*_{NS}(n) = C_{NS}(n)-C^*(n)$ of optimal allocation over no sharing.

Partial Variable List

n: Number of homogenous sources being multiplexed

$g(x)$: Effective bandwidth for a data source as a function of buffer length x.

C_B: Cost per unit buffer

C_C: Cost per unit bandwidth or service rate.

m: Ratio of C_B / C_C.

$P^*(n)$, $P_{FB}(n)$, $P_{IB}(n)$, $P_{NS}(n)$: Optimal Allocation, Fixed Buffer, Incremented Buffer, and No Share buffer and bandwidth allocation paths for n multiplexed sources.

$C^*(n)$, $C_{FB}(n)$, $C_{IB}(n)$, $C_{NS}(n)$: Cost of Optimal Allocation, Fixed Buffer, Incremented Buffer, and No Share combination.

$x^*(n)$: Optimum buffer allocation for n multiplexed sources.

$n^*(x)$: The inverse function of $x^*(n)$.

$\Delta C^*_{FB}(n)$, $\Delta C^*_{IB}(n)$, $\Delta C^*_{NS}(n)$: Additional cost of $C_{FB}(n)$, $C_{IB}(n)$, $C_{NS}(n)$ and over $C^*(n)$.

III. General Sources

In this section, we are concerned with results for the wide category of data sources for which effective bandwidth functions are well behaved convex, decreasing functions of buffer given a maximum loss criteria. We start by examining the conditions of optimal allocation assignment. We will show that the optimal buffer allocation $x^*(n)$ is strictly increasing with the number of sources n. We obtain an expression for the relationship of bandwidth to buffer when optimal allocation is achieved and we will show that this relationship can be written as a product of the cost ratio m and a function of the effective bandwidth $g(x)$. We will finish this section by showing that the cost savings, $\Delta C^*_{FB}(n)$ and $\Delta C^*_{NS}(n)$, are both convex and increasing with the number of sources n.

Consider the effective bandwidth function $g(x)$ in Figure 2. We expect $g(0)$ to be equal or close to the peak rate of the source and $g(x)$ to approach the mean rate as x approaches infinity. We impose a broad assumption that $g(x)$ is a continuous, decreasing function of x, $x \in [0, \infty)$.

By (EQ 1), it follows that the equal cost line is tangent to the $n\, g(x^*(n))$ curve as in (EQ 10). If this condition can not be met, then $x^*(n) = 0$.

$$n \cdot \frac{dg}{dx^*} = -m \qquad x^*(n) > 0 \tag{EQ 10}$$

For non-zero *x*, we may reverse (EQ 10) to obtain the number of sources which can be accommodated under the optimality condition. In this analysis, we allow *n*(x)* to be real valued.

$$n^*(x) = \frac{-m}{\frac{d}{dx}g(x)} \qquad x > 0 \tag{EQ 11}$$

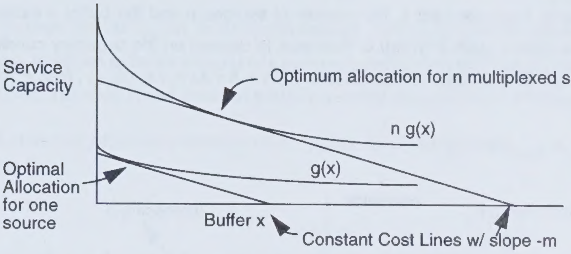

Service Capacity
Optimum allocation for n multiplexed sources
n g(x)
Optimal Allocation for one source
g(x)
Buffer x
Constant Cost Lines w/ slope -m

Figure 2 Minimum cost conditions

We now show that the optimal buffer *x*(n)* is strictly increasing with n in the range where *x*(n)* >0.

Theorem 1:

An optimal buffer assignment *x*(n)* which solves the minimization problem in (EQ 1) is strictly increasing with the number of sources *n* for *x*(n)* >0.

proof:

Suppose $x^*(n) \geq x^*(n+1)$ then

$$n\frac{d}{dx}g(x^*(n)) = (n+1)\frac{d}{dx}g(x^*(n+1)) = -m$$

$$\frac{\frac{d}{dx}g(x^*(n+1))}{\frac{d}{dx}g(x^*(n))} = \frac{n}{n+1} < 1$$

However, if $x^*(n) \geq x^*(n+1)$ then

$$\frac{\frac{d}{dx}g(x^*(n+1))}{\frac{d}{dx}g(x^*(n))} \geq 1$$

since *dg(x)/dx* is negative and strictly decreasing in magnitude. The contradiction implies x*(n) < x*(n+1).

Next, we consider the path of optimal buffer and bandwidth allocation *P*(n)*. By combining (EQ 2) and (EQ 11), we are able to eliminate the parametric variable *n* and show the relationship of bandwidth to buffer. Note that the capacity resource for a given buffer *x* is the ratio of the effective bandwidth to the magnitude of its derivative scaled by the cost ratio *m*.

$$P^*(n(x)) = \left[x(n), \frac{-m}{\frac{dg}{dx}} g(x(n)) \right] = \left[x, \frac{mg(x)}{\left|\frac{dg}{dx}\right|} \right] \qquad \forall x \geq x^*(1) \qquad \text{(EQ 12)}$$

Figure 3 illustrates an optimal path. From theorem 1, the number of sources *n* and the buffer *x* increase together. The valid domain of the optimal path *P*(n*(x))* is illustrated to depend on the boundary condition *x*(1)*. It can be shown that if *g(x)* has an algebraic, asymptotic form $g(x) = B + Ax^{-\nu}, \nu \in (0, \infty)$, then *P*(n(x))* will be convex and increasing.

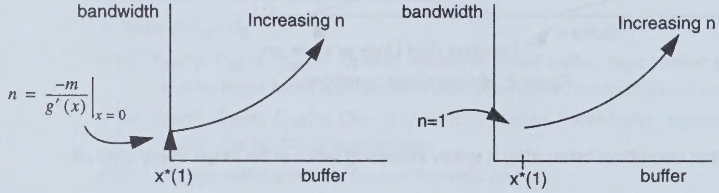


Figure 3 Boundary conditions on optimal path of buffer and bandwidth allocation.

The optimal allocation paths above show how resources increase along the path of minimum cost *P*(n)*. We compare this minimum cost path with the cost paths of the fixed buffer allocation and the no sharing allocation. In the following 2 theorems, we show the cost savings to be increasing convex with n.

Theorem 2:

The cost saving $\Delta C^*_{FB}(n)$ of optimal allocation over fixed buffer allocation is i) increasing with *n* and ii) convex in n for n such that *x*(n)*>0.

proof:

Figure 4 illustrates the effective bandwidth curves for *n, n+1, and n+2* multiplexed sources. The points labeled *P*(n), P*(n+1),* and *P*(n+2)* are optimal resource allocation points for *n, n+1*, and *n+2* sources respectively and the slope of the curve tangent to each of these points is identically *-m* by the optimality condition. The tangent lines shown may be considered as constant cost lines. The distances labeled Δ_n, Δ_{n+1}, and Δ_{n+2} are proportional to the extra cost of the *Fixed Buffer* scheme over the optimal allocation points respectively since each point on the tangent lines shown have equal costs. We recognize that the additional cost $\Delta C^*_{FB}(n)$ is equal to $C_C \Delta_n$. Formulate Δ_n, Δ_{n+1}, and Δ_{n+2} as follows:

$$\Delta_n = \int_{x^*(1)}^{x^*(n)} \left[n\left[-\frac{\partial}{\partial x}(g(x))\right] - m\right] dx = \int_{x^*(1)}^{x^*(n)} n|g'(x)| - mdx$$

$$\Delta_{n+1} = \int_{x^*(1)}^{x^*(n)} (n+1)|g'(x)| - mdx + \int_{x^*(n)}^{x^*(n+1)} (n+1)|g'(x)| - mdx$$

$$\Delta_{n+2} = \int_{x^*(1)}^{x^*(n)} (n+2)|g'(x)| - mdx + \int_{x^*(n)}^{x^*(n+1)} (n+2)|g'(x)| - mdx + \int_{x^*(n+1)}^{x^*(n+2)} (n+2)|g'(x)| - mdx$$

To show that $\Delta C^*_{FB}(n)$ is strictly increasing, we simply take the difference of Δ_{n+1}-Δ_n. The result can be shown to be the integral of a positive quantity over the positive distance from *x*(n)* to *x*(n+1)* (from Theorem 1) and a second positive quantity integrated over *x*(1)* to *x*(n)*.

Let us now prove the convexity of $\Delta C^*_{FB}(n)$ by showing that (Δ_{n+2}-Δ_{n+1})-(Δ_{n+1}-Δ_n) is positive:

$$(\Delta_{n+2} - \Delta_{n+1}) - (\Delta_{n+1} - \Delta_n) = \int_{x^*(n)}^{x^*(n+1)} m - n|g'(x)|dx + \int_{x^*(n+1)}^{x^*(n+2)} (n+2)|g'(x)| - mdx \qquad \text{(EQ 13)}$$

If both integrals are positive then the sum of the two is indeed positive. Let us eliminate *m* from the above integrals by using the optimality condition from (EQ 10) as specified below:

$$\begin{aligned} m &= n|g'(x^*(n))| \\ m &= (n+2)|g'(x^*(n+2))| \end{aligned} \qquad \text{(EQ 14)}$$

The first integral takes the following form:

$$\int_{x^*(n)}^{x^*(n+1)} m - n|g'(x)|dx = n \int_{x^*(n)}^{x^*(n+1)} |g'(x^*(n))| - |g'(x)|dx$$

In the range of *x* between *x*(n)* to *x*(n+1)*, the value of the integrated quantity is positive since the magnitude of the derivative $|g'(x)|$ is decreasing in value by the convex, decreasing assumption on *g(x)* and Theorem 1. Hence, the integral is positive.

The second integral in (EQ 13) is also found to be positive by substitution of the second line in (EQ 14):

$$\int_{x^*(n+1)}^{x^*(n+2)} (n+2)|g'(x)| - mdx = (n+2) \int_{x^*(n+1)}^{x^*(n+2)} |g'(x)| - |g'(x^*(n+2))|dx$$

In this case $|g'(x)|$ is greater or equal with $|g'(x^*(n+2))|$ in the range of *x* between *x*(n+1)* and *x*(n+2)* by Theorem 1 and the assumed convexity of *g(x)*.

Therefore (Δ_{n+2}-Δ_{n+1})-(Δ_{n+1}-Δ_n) is positive for *n* which satisfy the optimality conditions, i.e. *x*(n)>0* and $n \geq 1$.

Theorem 3:

The cost saving $\Delta C^*_{NS}(n)$ of optimal allocation over no sharing allocation is i) increasing with n and ii) convex in n for n such that $x^*(n)>0$.

proof:

From (EQ 4) and (EQ 8), we see that the additional resource of no sharing over fixed buffer allocation is $(n-1)x^*(1)$ with a linearly scaling cost of $C_B(n-1)x^*(1)$. Thus, our quantity of interest, $\Delta C^*_{NS}(n) = \Delta C^*_{FB}(n) + C_B(n-1)x^*(1)$. $\Delta C^*_{FB}(n)$ is increasing and convex from Theorem 2. The additive term $C_B(n-1)x^*(1)$ is linearly increasing with n. Therefore, $\Delta C^*_{NS}(n)$ is also increasing in and convex in *n* with the same restrictions for *n* as given in Theorem 2.

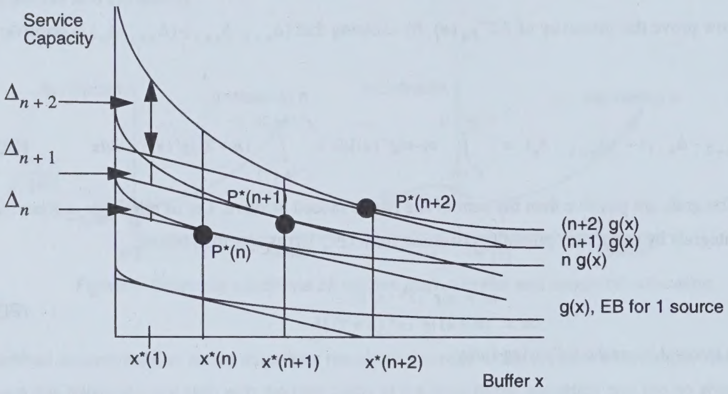


Figure 4 Illustration of cost savings of optimal allocation over fixed bandwidth allocation.

IV. On/Off Markov Fluids Sources

In this section, we proceed to give expressions for quantities of interest including optimal path *P*(n(x))*, optimal cost path *C*(n)*, and cost savings $\Delta C^*_{FB}(n)$, $\Delta C^*_{IB}(n)$, and $\Delta C^*_{NS}(n)$ specific to the case of On/Off Markov fluid sources.

Consider this alternate expression for effective bandwidth $g_{On/Off}(x)$ in terms of buffer length where λ is the on rate [Elwa93]:

$$g_{\mathrm{On/Off}}(x) = a - bx + \sqrt{b^2x^2 + cx + a^2} \qquad a = \frac{\lambda}{2} \qquad b = \frac{\alpha+\beta}{2\gamma} \qquad c = \frac{(\alpha-\beta)\lambda}{2\gamma}$$

$$Generator = \begin{bmatrix} -\alpha & \alpha \\ \beta & -\beta \end{bmatrix} \qquad Rate = \begin{bmatrix} 0 \\ \lambda \end{bmatrix} \qquad \gamma = |ln(pb)| \tag{EQ 15}$$

Note, the effective bandwidth with no buffer is simply $g_{On/Off}(0) = 2a = \lambda$. The limiting value of *g(x)* as $x \to \infty$ is the mean rate of the source, $\lambda\left(\frac{\alpha}{\alpha+\beta}\right)$.

The optimal path for the On/Off source case is then found by substituting (EQ 15) into *P*(n(x))* in (EQ 12).

$$P^*_{\text{On/Off}}(n(x)) = \left[x,\ m\frac{(a - bx + \sqrt{b^2x^2 + cx + a^2})}{\left| \frac{b^2x + c/2}{\sqrt{b^2x^2 + cx + a^2}} - b \right|} \right] \qquad \text{(EQ 16)}$$

Now, we introduce a Taylor Series approximation for the effective bandwidth function so that we might characterize the nature of the optimal path, the multiplexing gains, and their respective sensitivities. Consider the following Taylor Series expansion of $g_{\text{ON/Off}}(x)$:

$$g_{\text{On/Off}}(x) = k_0 + \frac{k_1}{x} + \frac{k_2}{x^2} + O(x^{-3})$$

$$k_0 = \frac{\lambda\alpha}{\alpha + \beta} \qquad k_1 = \frac{\lambda^2\gamma}{4(\alpha + \beta)}\left[1 - \left(\frac{\alpha - \beta}{\alpha + \beta}\right)^2\right] \qquad k_2 = -\lambda^3\gamma^2\frac{(\alpha - \beta)}{4(\alpha + \beta)^3}\left[1 - \left(\frac{\alpha - \beta}{\alpha + \beta}\right)^2\right] \qquad \text{(EQ 17)}$$

We find the effective bandwidth to be decreasing to the mean rate k_0 with a dominant factor of k_1/x as *x* grows to infinity. It can be shown that k_0 and k_1 are both positive quantities. Using this approximation, $P^*(n(x))_{\text{On/Off}}$. can be expressed as:

$$P^*_{\text{On/Off}}(n(x)) = \left[x(n), \frac{-m}{\frac{dg}{dx}} g(x(n)) \right] = \left[x, m\frac{k_0}{k_1}x^2 + O(x) \right] \qquad \text{(EQ 18)}$$

We see that the capacity requirement is increasing with the square of buffer length which is consistent with illustration of a convex, increasing optimal allocation path (Figure 1).

The cost savings $\Delta C^*_{FB,\text{ On/Off}}(n)$, $\Delta C^*_{IB,\text{ On/Off}}(n)$, and $\Delta C^*_{NS,\text{ On/Off}}(n)$ of optimal allocation over Fixed Buffer allocation, Incremented Buffer allocation, and no sharing respectively for the case of On/Off Markov fluid sources can be found from (EQ 2) - (EQ 9), (EQ 17), and (EQ 18).

$$\Delta C^*_{FB,\text{ On/Off}}(n) = nC_C[g(x^*(1)) - k_0] - 2\sqrt{n}\sqrt{C_B C_C k_1} + O(1) \qquad \text{(EQ 19)}$$

$$\Delta C^*_{IB,\text{ On/Off}}(n) = nC_B(x^*(1)) - 2\sqrt{n}\sqrt{C_B C_C k_1} + O(1) \qquad \text{(EQ 20)}$$

$$\Delta C^*_{NS,\text{ On/Off}}(n) = n[C_C(g(x^*(1)) - k_0) + C_B x^*(1)] - 2\sqrt{n}\sqrt{C_B C_C k_1} + O(1) \qquad \text{(EQ 21)}$$

Each of the multiplexing gains have a form of $c_0 n - c_1\sqrt{n} + O(1)$ and agree with the previous results concerning convexity of cost savings. The savings over the Fixed Buffer scheme is increasing with a dominant term of *n* times *(g(x*(1))* minus the mean source rate scaled by the bandwidth cost C_C. The savings over the Incremented Buffer scheme has a dominant term of *n* times the optimal buffer value *x*(1)* scaled by the buffer cost C_B. Finally, the savings over the No Sharing Scheme shows its dominant value term to be the sum of the previous two dominant term values.

V. Summary

In this work, we have introduced the concept of an increasing, convex optimal resource path of buffer and bandwidth parameterized by the number of homogenous sources being multiplexed. The gains of choosing optimal resource allocations due to temporal gains in multiplexing have been shown to be convex, increasing functions of the number of sources being multiplexed over certain alternative allocation schemes. For Markov Fluid On/Off sources, the cost savings is shown to have a form of $c_0 n - c_1 \sqrt{n} + O(1)$. This work is applicable to situations where resources in a switch or multiplexing system must be partitioned among different source classes. A study of this phenomena which incorporates the non-temporal gains in multiplexing would be a natural extension and future direction.

VI. References

[1] A. I. Elwalid and D. Mitra, “Analysis and design of rate based congestion control of high speed networks, I: stochastic fluid models, access regulations,” *Queueing Systems* 9 pp. 29-64, 1991.

[2] A. I. Elwalid, D. Mitra, “Effective Bandwidths of General Markovian Traffic Sources and Admission Control of High Speed Networks”, *IEEE/ACM Transactions in Networking, Vol. 1, No. 3,* pp. 329-343, 1993.

[3] R.J. Gibbens and P.J. Hunt, “Effective bandwidths for the multi-type UAS channel,” *Queueing Systems* 9 pp. 17-28, 1991.

[4] Roch Guerin, Hamid Ahmadi, and Mahmoud Naghshineh, “Equivalent Capacity and Its Application to Bandwidth Allocation in High Speed Network,” *IEEE JSAC* Vol. 9, No. 7, pp. 968-981, 1991.

[5] G. Kesidis, J. Walrand, and C.-S. Chang, “Effective Bandwidths for Multiclass Markov Fluids and Other ATM Sources, *IEEE ACM / Transactions on Networking*, Vol. 1, No. 4. pp. 424-428, August 1993.

[6] K. Rege, “Equivalent Bandwidth and Related Admission Criteria for ATM systems - A Performance Study,” *International Journal of Analog and Digital Communication Systems*, Vol 7, July, pp.181-191, 1994.

[7] I. Sidhu and S. Jordan, “Multiplexing Gains in Bitstream Multiplexors,” To appear in *IEEE / Transactions on Networking.*

Progressive Image Transmission on a Channel with Memory

B. S. Srinivas Eve A. Riskin Richard Ladner
Murat Azizõglu
University of Washington, Seattle

Abstract

We present results for progressive image transmission on a burst error channel modelled as a Gilbert channel. Interleaving followed by a Maximum A Posteriori (MAP) detector is used. The proposed MAP detector exploits the memory of the channel and the spatial correlation of the data to give a 1.5 to 2 dB improvement in the SNR over the MAP detector designed for a memoryless channel over a wide range of channel error rates.

1 Introduction

Data compression is the conversion of one representation of information into another at a lower bit rate. Compression mitigates the problems of limited storage space and transmission bandwidth. Vector quantization (VQ), a popular lossy compression technique, has been used extensively for this purpose [1].

Progressive transmission of images enables rapid recognition of their content at the receiver as well as early interruption of transmission if the image being sent is not desired resulting in savings in bits/time. Full-search VQ encoded data can be transmitted progressively using a full-search VQ progressive transmission tree (henceforth, called a *PT tree*) designed using Principal Component Partitioning (PCP) [7]. PCP-based codebook organization also helps reduce the impact of noise on VQ [7].

Most data compression techniques reduce the bit rate by eliminating, to the extent possible, the redundancy in the source. However, the loss of this redundancy increases the vulnerability of the source to channel noise which can result in significant quality degradation. In many practical situations, delay/complexity constraints and imperfect knowledge of the source characteristics result in a suboptimal source coder. This suboptimality is reflected by the fact that the source code does not eliminate all the redundancy in the source.

The residual redundancy in the output bit stream (due to a suboptimal source coder) can be exploited by the receiver to correct channel errors without the expense of channel coding. A Maximum A Posteriori (MAP) receiver has been proposed in the past for this purpose [5]. Previous work has examined the impact of channel errors in memoryless channels (such as the Binary Symmetric Channel (BSC)) on VQ performance [5, 8]. A notable exception is recent work by Phamdo *et al.* [4] where they have considered the impact of bursty channel errors, in a Polya contagion urn based model, on a synthetic generalized Gaussian source. Channel optimized VQ and interleaving with memoryless MAP detection were considered.

This paper considers the impact of burst errors generated by a Gilbert channel on VQ distortion performance for progressive image transmission. The Gilbert channel is one of the simplest and commonly accepted models for a channel with memory. We study the improvement in performance achieved by using the memory of the channel. This paper, similar to Hagenauer's work on convolutional coding in the presence of channel memory [3], incorporates channel state information in the Viterbi algorithm used in the MAP detector.

2 Channel Model

In most practical channels, such as wireless channels, errors tend to occur in clusters separated by fairly long error-free gaps. Among the causes of this clustering behavior in channel errors are fading, natural phenomena (e.g. lightning strikes), and switching transients. "Memory" is the channel's tendency to generate clusters of error bits.

We consider a binary-input, binary-output channel with memory modelled by the Gilbert channel [2]. It is a two state model with a GOOD (**G**) and BAD or BURSTY (**B**) state (see Figure 1). The state model generates a sequence $\{Z_i\}$ of binary noise digits similar to typical error sequences generated by a real channel. In state **G**, the model produces only zero ($Z_i = 0$) noise digits corresponding to noise-free transmission. However, in state **B**, the value of the noise digit generated is determined probabilistically. In other words, in state **B**, $Z_i = 1$ with probability ϵ (and $Z_i = 0$ with probability $1-\epsilon$). After generating Z_i, the model makes a state transition before generating Z_{i+1}. Dense bursts are simulated by making the model's states persist over a length of time and $\epsilon \gg 0$. This is achieved by ensuring that the transition probabilities, $P_{GB} = \text{Prob}(\mathbf{G}\rightarrow\mathbf{B})$ and $P_{BG} = \text{Prob}(\mathbf{B}\rightarrow\mathbf{G})$, are small and, consequently, the self-transition probabilities $P_{GG} = \text{Prob}(\mathbf{G}\rightarrow\mathbf{G})$ and $P_{BB} = \text{Prob}(\mathbf{B}\rightarrow\mathbf{B})$, are large. Due to the Markovian nature of the state space, the run lengths of the states are geometrically distributed with mean lengths of $1/P_{GB}$ for the G-runs and $1/P_{BG}$ for the B-runs respectively. The average bit error rate of the channel is $\sigma = \epsilon\pi(BAD)$.

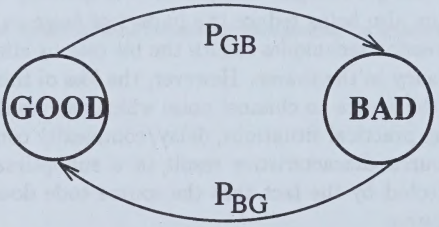


Figure 1: Gilbert channel model with state transition probabilities P_{GB}=Pr($\mathbf{G}\rightarrow\mathbf{B}$) and P_{BG}=Pr($\mathbf{B}\rightarrow\mathbf{G}$).

3 Interleaving

Burst error channels display significant correlation between bit error locations. The *channel memory length* is proportional to the expected sojourn time of the burst (**B**) state (average run length of the burst state). We use *interleaving* before our MAP detector since the Viterbi algorithm is sensitive to clustered errors. Interleaving requires no knowledge of channel memory statistics other than its approximate length [6]. It fragments and disperses the memory in the channel and renders it effectively memoryless. In other words, it eliminates the temporal correlation in the channel error process by scrambling a bit-plane such that spatially adjacent bits are not temporally adjacent in terms of channel transmissions (see Figure 2). The temporal separation between spatially adjacent bits must be greater than the channel memory length to eliminate correlation in their error status. This reduces the probability of spatially clustered errors. The *interleaving depth* (I) is defined as the spatial separation between temporally adjacent bits. The only drawback of this technique is the delay involved in interleaving.

We choose the depth of each sub-trellis to be D $(= \frac{N}{I})$ where N is the number of bits in a bit-plane. Let $\{X_i : 1 \leq i \leq N\}$ be a bit-plane of the VQ encoder's output bit stream. Also, let $\{T_i : 1 \leq i \leq N\}$ be the interleaved bit stream from 1 bit-plane transmitted on the channel. The interleaving operation can be described as a mapping from $\{X_i\}$ to $\{T_i\}$,

$$X_{I(k-1)+n} = T_{(n-1)D+k} \qquad (1)$$

where, $1 \leq n \leq I$ and $1 \leq k \leq D$. In Equation (1), k refers to the temporal sequence of the bits in one scan of the bit-plane while n refers to the scan number (each bit-plane is interleaved by scanning it I times).

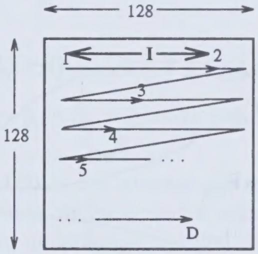


Figure 2: Interleaving pattern in a 128 × 128 bit-plane. D = the temporal separation between spatially adjacent bits, I = Interleaving Depth. The numbers represent the temporal order of transmission.

4 MAP Detection

We apply MAP detection after deinterleaving at the receiver. The proposed MAP detector, which exploits the memory in the Gilbert channel, uses a 3 × 3 mask (see Figure 3). Owing to the fragmentation of the channel memory due to interleaving, the

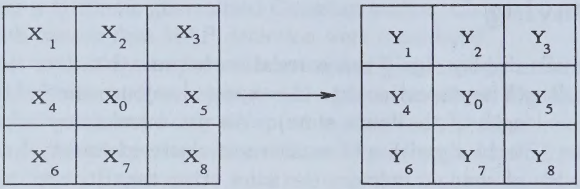


Figure 3: The 3×3 mask. X_i correspond to the transmitted bits while Y_i correspond to the received bits.

error status of the 8 neighboring received bits $(Y_1, Y_2, \ldots, Y_8)$ of each central received bit (Y_0) are assumed to be independent of each other as well as the central bit.

The two-state Gilbert model (see Figure 1) with BAD state error probability ϵ can be transformed into a three state model by splitting the BAD state into two states (B0 and B1) [2]. In the resulting three state model (see Figure 4), a zero noise bit $(Z_i = 0)$ is generated when the channel is either in states G or B0 while a one noise bit $(Z_i = 1)$ is generated only in the B1 state. The equilibrium state probabilities in the transformed three state model are:

$$\begin{aligned} \pi(G) &= \frac{P_{BG}}{P_{GB} + P_{BG}} \\ \pi(B0) &= \frac{(1-\epsilon)P_{GB}}{P_{GB} + P_{BG}} \\ \pi(B1) &= \frac{\epsilon P_{GB}}{P_{GB} + P_{BG}} \end{aligned} \tag{2}$$

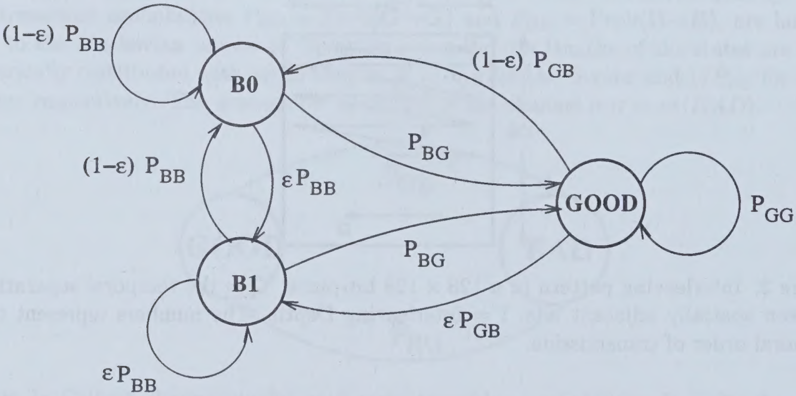


Figure 4: Three state Gilbert model. $P_{GG} = 1 - P_{GB}$ and $P_{BB} = 1 - P_{BG}$.

Memory in the Gilbert channel is exploited in the design of the proposed MAP detector by using a Viterbi-like construction. Each bit-plane is represented by a trellis, where each stage of the trellis represents a bit position in the bit-plane. We have found that the MAP detector works better when it uses the updated version of each bit when available. Consequently, to be able to incorporate the updated versions of bits in the proposed MAP detector, the trellis for the entire bit-plane is broken up into a number of sub-trellises. Each sub-trellis corresponds to the bits covered in one interleaving sweep of a bit-plane. Successive stages of a sub-trellis correspond to temporally adjacent bit positions during transmission. In other words, successive stages of the sub-trellis correspond to bits spatially separated by I.

Using the Viterbi algorithm and the received bit sequence ($\{Y_i\}$), the most likely D-step state sequence $\{\Sigma_i : 1 \leq i \leq D\}$ can be estimated which, in turn, determines the most likely error bits at the appropriate locations $\{Z_i : 1 \leq i \leq D\}$. Consequently, the most likely transmitted bit sequence $\{X_i : 1 \leq i \leq D\}$, at the appropriate locations in the bit-plane, can be determined by,

$$X_i = Z_i \oplus Y_i, \qquad \text{for} \qquad 1 \leq i \leq D \tag{3}$$

where $\oplus$ denotes binary addition.

The error status of the central bit in the 3 × 3 mask is determined by the channel state in the corresponding stage of the trellis. Each trellis stage has 3 states corresponding to the states of the transformed Gilbert model (see Figure 4). In states G and B0, X_0 is error-free while in the $B1$ state, X_0 is flipped. For computational simplicity (without significant loss in performance), it is assumed that at most one channel error can occur in the 8 neighboring bits in the mask.

The algorithm requires the computation of 3 state dependent metrics ($\Lambda(G), \Lambda(B0)$ and $\Lambda(B1)$). Each metric corresponds to the conditional probability of the received bit conditioned on the corresponding state of the channel (i.e. $\Lambda(\Sigma) = \Pr(Y_0 = y_0 \mid \Sigma)$). In other words, for each bit position or, equivalently, stage of the sub-trellis,

$$\left.\begin{array}{c}\Lambda(G)\\ \Lambda(B0)\end{array}\right\} = \left[\begin{array}{l}\Pr(X_i = y_i, 0 \leq i \leq 8)(1-\sigma)^8 \quad + \\ \displaystyle\sum_{i=1}^{8} \Pr(X_i = \overline{y}_i \wedge X_j = y_j, j \neq i, 0 \leq j \leq 8)(1-\sigma)^7\sigma\end{array}\right] \tag{4}$$

$$\Lambda(B1) \quad = \quad \left[\begin{array}{l}\Pr(X_0 = \overline{y}_0 \wedge X_i = y_i, 1 \leq i \leq 8)(1-\sigma)^8 \quad + \\ \displaystyle\sum_{i=1}^{8} \Pr(X_0 = \overline{y}_0 \wedge X_i = \overline{y}_i \wedge X_j = y_j, j \neq i, 1 \leq j \leq 8)(1-\sigma)^7\sigma\end{array}\right] \tag{5}$$

where $\sigma = \pi(B1) = \epsilon\pi(BAD)$ is the average bit error rate of the channel. As is standard in Viterbi formulations, we have used a negative logarithm to ensure that all quantities are positive and an underflow is avoided. Having computed the metrics for the current stage, the Viterbi algorithm then computes, for each state in that stage, the cumulative path length of the most probable state sequence (shortest *survivor*) terminating in the corresponding state. The cumulative path lengths terminating in the three states are denoted by $\Gamma(G), \Gamma(B0)$, and $\Gamma(B1)$ respectively. Given no information about the starting state of the first sub-trellis, the cumulative path lengths are initialized by, $\Gamma(\Sigma) = -\log(\Lambda(\Sigma)\pi(\Sigma))$, where the equilibrium state probabilities, π, are given by (2). However, for initializing the algorithm for subsequent sub-trellises, $\pi(\Sigma_1^{k+1})$ ($\Sigma_1^{k+1} \in \{G, B0, B1\}$) is replaced by $\Pr(\Sigma_1^{k+1} \mid \Sigma_D^k)$ (where the superscript refers to the sub-trellis number and the subscript refers to the stage number in the sub-trellis) given in Figure 4. Here, Σ_D^k refers to the most probable state in the last stage of the current sub-trellis as given by running the Viterbi algorithm on it. Note that the bit position corresponding to the last stage of the k^{th} sub-trellis is temporally adjacent to the bit corresponding to the first stage of the $(k+1)^{th}$ sub-trellis, thus providing justification for the substitution.

For the succeeding stages of the sub-trellis, to compute the cumulative path lengths for each state $\Sigma_{i+1}^k \in \{G, B0, B1\}$, branch metrics $(\lambda(\Sigma_{i+1}^k, \Sigma_i^k))$ are first computed,

$$\lambda(\Sigma_{i+1}^k, \Sigma_i^k) = -\ln(\Lambda(\Sigma_{i+1}^k)\Pr(\Sigma_{i+1}^k \mid \Sigma_i^k)) \qquad \forall \quad \Sigma_i^k \in \{G, B0, B1\} \tag{6}$$

The candidate cumulative path lengths are then computed by,

$$\Gamma(\Sigma_{i+1}^k, \Sigma_i^k) \;=\; \Gamma(\Sigma_i^k) + \lambda(\Sigma_{i+1}^k, \Sigma_i^k) \qquad \forall \quad \Sigma_i^k \in \{G, B0, B1\} \tag{7}$$

Finally, the updated cumulative path length for each state $\Sigma_{i+1}^k \in \{G, B0, B1\}$ is given by,

$$\Gamma(\Sigma_{i+1}^k) \;=\; \min_{\Sigma_i^k}(\Gamma(\Sigma_{i+1}^k, \Sigma_i^k)) \tag{8}$$

This procedure is repeated for succeeding stages of the sub-trellis. On reaching the last stage, the trellis state corresponding to the shortest path (smallest Γ) represents the final state and the survivor path from the first stage of the sub-trellis to the final state in the last stage gives the most probable state sequence.

5 Results and Discussion

A training sequence is used for designing the PCP-based VQ codebook as well as to determine the joint source probabilities used in Equations (4) and (5). The training sequence consisted of 20 magnetic resonance (MR) images. The test sequence was a set of 5 MR images distinct from the training set. To achieve progressive transmission, the input image (9 bpp) was first encoded to 2 bpp using a size 256 4-dimensional PCP-organized VQ.

In our experiments, we fixed $\epsilon = 0.5$ and $P_{BG} = 0.02$ and varied the average error rate by varying P_{GB}. In other words, the duration of the GOOD bursts were varied while the other parameters were kept fixed. An interleaving depth of 101 was chosen to optimize the sub-trellis depth for the Viterbi algorithm.

The decreasing correlation in a bit-plane (see Figure 5) results in the MAP detector being useful only for the first three bit-planes (MSB, MSB-1 and MSB-2). Note further that the decreasing trend in correlation among the first three bit-planes results in the MAP detector (with memory as well as memoryless [7]) being most effective for the MSB bit-plane and decreasingly effective for the others (see Figure 6(a)). Hence, the MAP detector is used only for the first 3 bit-planes. However, due to the PCP-based codebook organization, the less significant bits have a smaller impact on average distortion [7] and the inability to carry out MAP detection for the later bits of a codeword does not lead to a significant increase in distortion.

The performance gain achieved by exploiting the memory of the channel may be described in several ways. As seen in Figure 6(a), the proposed MAP detector outperforms a memoryless MAP detector [7] over a range of average channel error rates. This fact is further reinforced by examining Figure 6(b) which is a plot of the overall SNR (using the recomposed codewords at the receiver after receiving all the bit-planes) versus the average bit error rate for both MAP detectors. Once again, the plot confirms

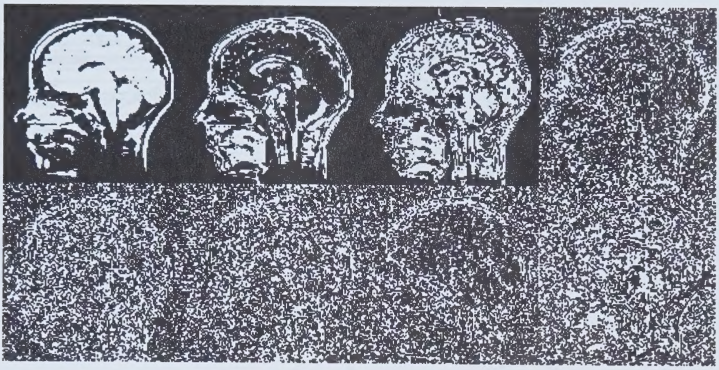
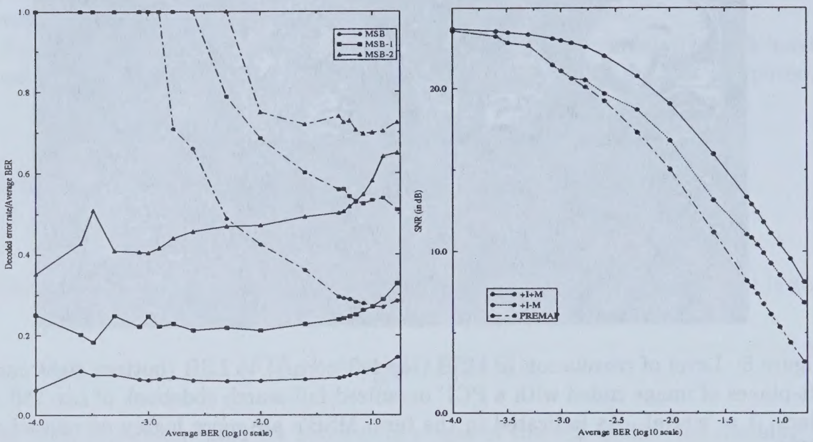


Figure 5: Level of correlation in MSB (top left corner) to LSB (bottom right corner) bit-planes of image coded with a PCP organized full-search codebook of size 256 (0 = black, 1 = white). As indicated in the text, MSB's are more highly correlated than LSB's.

the superior performance achieved by making use of the channel memory as against ignoring it. Taking advantage of the channel memory, over a wide range of channel error rates (10^{-4}–10^{-1}), the proposed MAP detector achieves an SNR gain of 1.5 to 2 dB over memoryless MAP which, in turn, gives a 2.5 dB increase in SNR over no postprocessing at all (i.e. no MAP detection).

Furthermore, the ability of the proposed MAP detector to successfully predict the channel state is depicted in Figure 7. The streaks observed in Figure 7(a) correspond to the time periods during which the channel is in the BURST state (average BER = 10 %). On the other hand, the streaks in Figure 7(b) represent the temporal locations of the BURST states as estimated by the MAP detector. The mistakes made by the proposed MAP detector in estimating the true channel state at any point in time is shown in Figure 7(c). Since, in the Gilbert channel model, both noisy and error-free digits are generated in the BURST state, the temporal locations of the noise digits ($Z_i = 1$) are also displayed as in Figure 7(d). The error streaks (seen in Figure 7(d)), in turn, are dispersed by interleaving as seen in Figure 7(e) resembling the error pattern of a BSC. As just discussed, the streaks in Figures 7(a) and 7(b) represent the temporal locations of the BURST states and not the actual error locations *in* an image bitplane during transmission. Instead, they represent the fact that once the channel enters a state it tends to stay in that state. As a result of using a MAP detector, errors might remain uncorrected or be newly introduced. The ability of the proposed MAP detector, in tandem with interleaving, to correct most of the errors in the MSB bit-plane can be seen in Figure 7(f). Note that the error pattern after MAP detection resembles the brain image used in these experiments. This is most likely due to the difficulty of correcting errors occuring at edges in the images.

The error correcting capability of the proposed MAP detector in comparison to that of a memoryless MAP is presented in Figure 8. This series of images provides a more dramatic demonstration of the improvement in error correction with the proposed MAP detector.



(a) (b)

Figure 6: (a) Error rate improvement using MAP with memory (solid line) and memoryless MAP (dashed line) for the first 3 bit-planes (MSB,MSB-1,MSB-2), (b) SNR improvement using MAP with memory. Using the channel memory gives approximately 2 dB SNR improvement over using memoryless MAP. +I+M = interleaving and MAP with memory, +I-M = interleaving and memoryless MAP, PREMAP = no postprocessing.

6 Conclusion

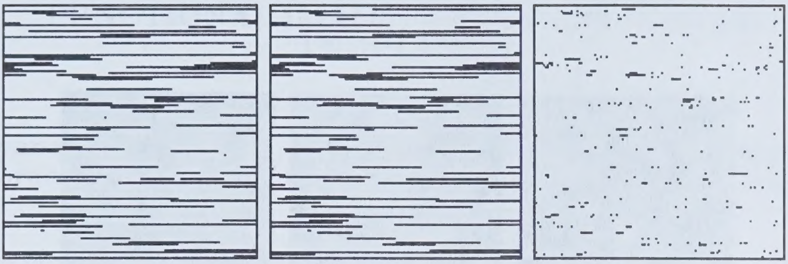
We investigated the improved performance of a MAP detector achieved by exploiting the memory of a burst error channel, the Gilbert channel. Progressive transmission of a real image compressed using a PCP-organized VQ codebook was studied. An SNR gain of about 2 dB was attained by the MAP detector designed for the Gilbert channel in comparison to a memoryless MAP. Thus, we have been able to demonstrate that temporal correlation of channel errors and spatial correlation of data bits in an image can be incorporated into a detection algorithm which outperforms other approaches that utilize only the latter information.

References

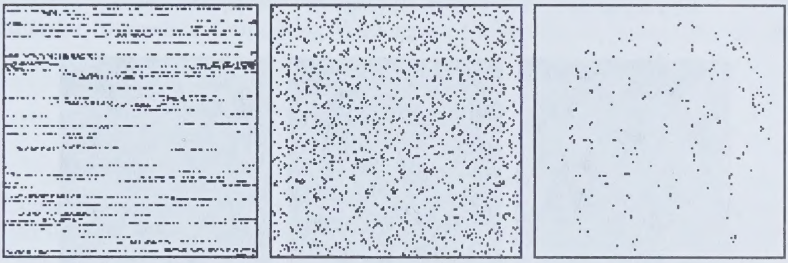
[1] H. Abut, editor. *Vector Quantization.* IEEE Press, Piscataway, NJ, May 1990.

[2] E. N. Gilbert. Capacity of a burst-noise channel. *Bell Syst. Tech. J.*, 39:1253–1266, Sept 1960.

[3] J. Hagenauer. Viterbi Decoding of Convolutional codes for Fading- and Burst-Channels. In *International Zurich Seminar on Digital Communications*, pages G2.1–G2.7, Mar 1980.



(a) G=white, B=black (b) G=white, B=black (c) CS=white, WS=black



(d) NE=white, E=black (e) NE=white, E=black (f) NE=white, E=black

Figure 7: (a) State pattern for MSB bit-plane showing the Gilbert channel burst state sequence with G = GOOD and B = BURST state [Average BER = 10%], (b) State sequence estimated by MAP detector, (c) States mislabelled by MAP with CS = correct state and WS = wrong state, (d) Error pattern before MAP for MSB bit-plane with NE = no error bit and E = error bit (here $\Phi(G) = \Phi(B0) = 0$ and $\Phi(B1) = 1$) (e) Error scattering resulting from interleaving before MAP, (f) Uncorrected and newly introduced errors due to MAP detection.

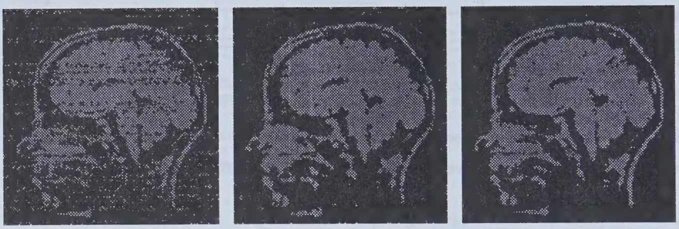
[4] N. Phamdo, F. Alajaji, N. Farvardin, and T. E. Fuja. Quantization of Memoryless Sources over Discrete Markov Channels. In *Proceedings of CISS*, pages 1–6, Mar 1994.

[5] N. Phamdo and N. Farvardin. Optimal detection of discrete Markov sources over discrete memoryless channels - applications to combined source-channel coding. *IEEE Trans. Inform. Theory*, IT-40(1):186–93, Jan 1994.

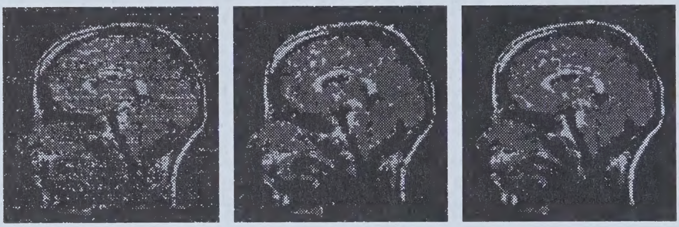
[6] A. J. Viterbi and J. K. Omura. *Principles of Digital Communication and Coding.* McGraw-Hill Book Company, 1979.

[7] R.-Y. Wang. *Organization of Fixed Rate Vector Quantization Codebooks.* PhD thesis, University of Washington, Seattle, WA, Jul 1994.

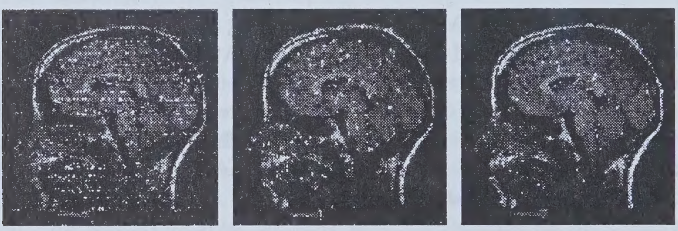
[8] R.-Y. Wang, E. A. Riskin, and R. Ladner. Codebook Organization to Enhance Maximum A Posteriori Detection of Progressive Transmission of Vector Quantized Images over Noisy Channels. To appear in *IEEE Trans. on Image Proc.*, Jan 1996.



(a) 1 bit (noisy) (b) 1 bit (map) (c) 1 bit (mmap)



(d) 2 bits (noisy) (e) 2 bits (map) (f) 2 bits (mmap)



(g) 8 bits (noisy) (h) 8 bits (map) (i) 8 bits (mmap)

Figure 8: Progressive Transmission of a 256 × 256 medical image over a bursty error channel with average BER = 10 %. (map = memoryless MAP, mmap = MAP with memory). The figures labeled (a), (d) and (g) are the noisy uncorrected images generated by sending 1, 2 and 8 bit-planes respectively. Figures (b), (e) and (h) are those recovered by using interleaving and memoryless MAP detection. Finally, figures (c), (f) and (i) are those recovered by using interleaving and MAP detection tailored to the Gilbert channel. The image (h) has a 2.5 dB increase in SNR over the image (g) while (i) has an additional 2 dB improvement in SNR over that of (h).

ACCOMMODATING ANATOMICAL VARIABILITY IN FUNCTIONAL IMAGING VIA DEFORMABLE TEMPLATES

Ayananshu Banerjee[1], Gary. E. Christensen[2,3], John Haller[3,4], Sarang Joshi[1,5], Marcus E. Raichle[3] and Michael I. Miller[1,3,5]

Department of Electrical Engineering, Washington University, St. Louis, MO 63130[1]
Department of Surgery, Washington Univ. School of Medicine, St. Louis, MO 63110[2]
Department of Radiology, Washington Univ. School of Medicine, St. Louis, MO 63110[3]
Department of Psychiatry, Washington Univ. School of Medicine, St. Louis, MO 63110[4]
Institute for Biomedical Computing Washington University, St. Louis, MO 63110[5]

Abstract

Mathematical techniques are presented for registering anatomical and functional modalities of different subjects via a vector field transformation applied to the underlying coordinate system of the anatomical template which maps the template image to the target image. A coarse-to-fine approach is adopted where the final high dimensional transformation is pre-conditioned by the lower dimensional information inserted by the anatomists. Transformations are constrained to be consistent with the physical properties of deformable elastic solids. After finding the transformation using the high-resolution anatomical data, it is applied to the functional modality image of the template to map it to that of the target. Results are presented in this paper that demonstrate this procedure on MRI and PET data.

1 Introduction

In today's scientific world, information regarding the anatomy and physiological function of animals and humans is easily obtained from different imaging modalities such as magnetic resonance imaging (**MRI**), X-ray computed tomography (**CT**) and positron emission tomography (**PET**). However our inability to relate such information to specific anatomical regions hinders our interpretation of the data. Hence registration of images obtained from a single subject using different modalities is becoming an increasingly useful technique for both research and patient care. We however are interested in inter-subject registration of different modalities and remove the anatomical variabilities among them.

Our motivation in this work is to map the PET data of one subject to the PET data of another. The common method for increasing signal to noise in PET images is by image averaging. In fact, single subjects studied in an activation paradigm may not produce a detectable signal in a subtraction image, where a baseline image is subtracted from an activation image. For this reason, it is often necessary to average multiple subjects' scans to create an average or statistical image of the activated state.

Transformation of the PET images to a standard template (atlas) space allows one to average multiple subjects scanned under identical conditions [4]. Image transformations of this kind may also be applied to comparisons between groups and to other functional imaging modalities (e.g. fMRI). But the PET data, which is very low resolution cannot be independently used in a efficient way to get these transformations. Therefore, we use high resolution MRI data to find the anatomical correspondence between two anatomies and then use the transformation to map the template PET (preregistered to its MRI image by the Wood et al. algorithm [5]) to the study PET image.

2 Anatomical atlas

As in [1], an ideal neuroanatomy atlas corresponds to an anatomical template which, in our case, is a multi-valued vector function on the ideal coordinates of the template. Variation is accommodated by defining probabilistic transformations applied to the templates. Mathematically we define the atlas to be a vector mapping to the coordinate system $\Omega \subset R^3$ according to $\mathbf{T} : \Omega \rightarrow \mathcal{T}$, with the range space $\mathcal{T}$, assumed to be an M-fold product of spaces $\mathcal{T}_1 \times \mathcal{T}_2 \times ... \mathcal{T}_M$, and each component $T_m \in \mathcal{T}_m$ corresponds to a different image in the atlas. For example $\mathcal{T}_1$ is MRI image, $\mathcal{T}_2$ is PET, $\mathcal{T}_3$ is CT and so on. The triplet $(\Omega, \mathbf{T}, \mathcal{T})$ is termed the *anatomical atlas.*

We attach to the template the Euclidian groups: a template corresponds to the equivalent class corresponding to the orbit of $(\Omega, \mathbf{T}, \mathcal{T})$ under group action. To account for the rigid global transformation we can write the transformation as

$$\vec{h} : \vec{x} \rightarrow \quad (A\vec{x} + b) \tag{1}$$

where b is a 3×1 column vector representing the translation vector and $A \in \mathbf{GL}(n)$, where $\mathbf{GL}(n)$ denotes the general linear group i.e. the group of all 3×3 matrix with determinant nonzero (invertible). By polar decomposition the non-singular matrix A can be written as $A = SO$, where S is a 3×3 positive definite symmetric matrix representing the *scale skewing matrix* and O is a 3×3 orthogonal matrix representing the *rigid rotation matrix.* The coefficients of the matrix A and vector b are determined by minimizing

$$\sum_{i=0}^{N} |\vec{y}_i - (A\vec{x}_i + b)|^2$$

where $\vec{x}_i$ are points on the template coordinate system and $\vec{y}_i$ are the corresponding points in the target coordinate system.

3 Transformations

We focus on the set of transformations $\vec{h} \in \mathcal{H}$ on the ideal coordinate system, where $\mathcal{H}$ is the set of maps $\vec{h} : \Omega \rightarrow \Omega$. These are constructed via a hierarchy of group maps $\vec{h}_i : \Omega \rightarrow \Omega$ defining the affine motion and are generated from product of the translation group applied to points $\vec{x} \in \Omega$:

$$\vec{h} : \vec{x} \rightarrow \vec{x} - \vec{u}(\vec{x}) \tag{2}$$

The vector field $\vec{u}(\vec{x}) = (u_1(\vec{x}), u_2(\vec{x}), u_3(\vec{x}))$ is called the *displacement field.* The maps constructed from this transformations allow the dilation, contraction and warping of the underlying coordinate system of the atlas into the coordinate system of the individual anatomy at a very local level. The set of anatomies generated from the atlas $(\Omega, \mathbf{T}, \mathcal{T})$ becomes $\{\mathbf{T} \circ \vec{h} : \vec{h} \in \mathcal{H}\}$, with $\circ$ the composition operator.

3.1 Manifold Transformation

Points, lines, surfaces and subvolumes corresponding to landmarks, sulcal lines, cortical surfaces and major nuclei are examples of 0, 1, 2, 3 dimensional manifolds. This information is provided by the anatomists and is used to generate low dimensional *manifold transformations.* We assume that we know the observable manifolds $\mathcal{M}(i)$ as well as the vector field on the manifold i.e. $\vec{u}(\vec{x}) = k(\vec{x}), \vec{x} \in \mathcal{M}(i)$ mapping the observable manifolds in the template to the subject.

The problem of determining the manifold transformation can be expressed as the minimization of $\int_\Omega |L\vec{u}(\vec{x})|^2 d\vec{x}$ subject to $u(\vec{x}) = k(\vec{x}), \vec{x} \in \bigcup_{i=0}^{3} \mathcal{M}(i)$, where L is a local (bounded support) self adjoint positive differential operator. We define $\vec{k}_i = \vec{y}_i - \vec{x}_i$, for $i = 1...N$ where $\vec{x}_i$ are N points on the template coordinate system and $\vec{y}_i$ are the corresponding points in the target coordinate system. For generality of the solution, we assume that the manifolds are collection of points and are not fully observable in the data. There are errors associated with the observation process, the errors being independent Gaussian distributed with zero mean and variance σ^2. This idea of constraining the maps via points, lines, surfaces or subvolumes can all be viewed as being the solution of a *generalized Dirichlet problem.* From [6] we can then write the Bayesian problem and its solution accordingly.

Given noisy observations, $u(\vec{x}_i) = \vec{k}_i = \hat{k}_i + \vec{n}_i$, *for* $i = 1...N$, *where* $\hat{k}_i$ *are constants and* $\vec{n}_i$ *are Gaussian distributed with mean zero and variance* σ^2. *The minimizer of*

$$\int_\Omega |L\vec{u}|^2 + \sum_{i=1}^{N} \frac{|\hat{k}_i - \vec{u}(\vec{x}_i)|^2}{\sigma^2}$$

satisfies

$$\vec{u}(\vec{x}) = \sum_{i=1}^{N} \beta_i K(\vec{x}, \vec{x}_i) \tag{3}$$

where $\beta = [\beta_i, ..., \beta_N]^t$ *satisfy the system of linear equations*

$$(K + \sigma^2 I)\beta = \hat{\mathrm{k}}$$

with $\hat{\mathrm{k}} = [\hat{k}_1, ...\hat{k}_N]^t$, *and* K,I *are* $N \times N$ *matrices with* $K(i,j) = K(\vec{x}_i, \vec{x}_j)$

3.2 Elastic Basis Transformation

We adopt a Bayesian approach for estimating the transformation $\vec{h}$ given the data $\{S(\vec{x}), \vec{x} \in \Omega\}$, which in our case is the MRI data and we take it to be a Gaussian random field with mean $(T \circ \vec{h})$. To estimate the transformation $\vec{h}$ that satisfies linear elastic-solid model (Miller *et al.* 1993) [1] Bayes rule

$$p(\vec{h}|S) \propto p(S|\vec{h})p(\vec{h})$$

is used to define the posterior in terms of a likelihood $p(S|\vec{h})$ and a prior model $p(\vec{h})$. Equivalently we can write this as

$$p(\vec{u}|S) \propto p(S|\vec{u})p(\vec{u}) \tag{4}$$

where the displacement field $\vec{u}(\vec{x})$ is given by

$$\vec{u}(\vec{x}) = \vec{x} - \vec{h}(\vec{x})$$

The *likelihood* models the sensor characteristics of the imaging system. We assume that the likelihood has a Gibbs' form $p(S|\vec{u}) = (1/Z)e^{-H(S|\vec{u})}$, where Z is a normalizing constant and $H(S|\vec{u})$ is the Gibbs' potential. In our case Gibbs' potential is the Gaussian potential given by

$$H(S|\vec{u}) = \frac{1}{2\sigma^2}\int_\Omega |T(\vec{x} - \vec{u}(\vec{x})) - S(\vec{x})|^2 d\vec{x} \tag{5}$$

where σ is a constant.

The *prior* $p(\vec{u})$ measures how likely the transformations are regardless of the study S. It ensures that the transformation $\vec{h} \in \mathcal{H}$ obeys the physical properties of elastic solids and is smooth and continuous. We assume that the prior has a Gibbs' form $(1/Z)e^{-H(\vec{u})}$ with Z as the normalizing constant and $H(\vec{u})$ as the prior potential given by linear small deformation elasticity.

$$H(\vec{u}) = \frac{\alpha}{2}\sum_{i=1}^{3}\sum_{j=1}^{3}\int_\Omega \lambda_0\left(\frac{\partial u_i(\vec{x})}{\partial x_i}\right)\left(\frac{\partial u_j(\vec{x})}{\partial x_j}\right) + \frac{\mu_0}{2}\left(\frac{\partial u_i(\vec{x})}{\partial x_j} + \frac{\partial u_j(\vec{x})}{\partial x_i}\right)^2 d\vec{x}\ , \tag{6}$$

where λ_0, and μ_0 are the Lame elasticity constants and α is a Lagrange multiplier.

It can be proved that the $\vec{u}(\vec{x})$ can be expressed in terms of a complete orthonormal basis such that

$$\vec{u}(\vec{x}) = \sum_{i,j,k=0}^{d}\sum_{n=1}^{3}\mu_{n,i,j,k}\vec{\phi}_{n,i,j,k}(\vec{x}) \tag{7}$$

and with the free boundary conditions [2, 3] these basis vectors are given by

$$\vec{\phi}_{1,i,j,k}(\vec{x}) = \alpha_1\begin{bmatrix} i\sin\pi i x_1\cos\pi j x_2\cos\pi k x_3 \\ j\cos\pi i x_1\sin\pi j x_2\cos\pi k x_3 \\ k\cos\pi i x_1\cos\pi j x_2\sin\pi k x_3 \end{bmatrix} \tag{8}$$

$$\vec{\phi}_{2,i,j,k}(\vec{x}) = \alpha_2\begin{bmatrix} -j\sin\pi i x_1\cos\pi j x_2\cos\pi k x_3 \\ i\cos\pi i x_1\sin\pi j x_2\cos\pi k x_3 \\ 0 \end{bmatrix}$$

$$\vec{\phi}_{3,i,j,k}(\vec{x}) = \alpha_3\begin{bmatrix} ik\sin\pi i x_1\cos\pi j x_2\cos\pi k x_3 \\ jk\cos\pi i x_1\sin\pi j x_2\cos\pi k x_3 \\ -(i^2+j^2)\cos\pi i x_1\cos\pi j x_2\sin\pi k x_3 \end{bmatrix}$$

The coefficients $\alpha_{n,i,j,k}$ give each eigenvector unit energy. We choose $\mu_{1,i,j,k}, \mu_{2,i,j,k}$ and $\mu_{3,i,j,k}$ of Equation(7) to be zero-mean, independent Gaussian random variables with variances $\sigma_{1,i,j,k} = 2/(\pi^2(\lambda_0 + 2\mu_0)(i^2 + j_2 + k_2))$ and $\sigma_{2,i,j,k} = \sigma_{3,i,j,k} = 2/(\pi^2\mu_0(i^2 +$

$j_2 + k_2)$) respectively. We have proved that this set of basis vectors $\{\phi_i\}$ are the eigen functions of the elasticity operator corresponding to

$$L\vec{u} = \mu_0 \nabla^2 \vec{u}(\vec{x}) + (\mu_0 + \lambda_0)\vec{\nabla}(\vec{\nabla} \cdot \vec{u}(\vec{x})) \tag{9}$$

Minimum mean squared error estimation

The conditional mean of the posterior which is also the minimum squared error (MMSE) estimate is used to determine the parameters of the optimal transformation that individualizes an atlas given a study. This is done by generating samples of the posterior distribution from the stochastic differential equation (SDE)

$$d\mu_{i,j,k,n}(t) = -\frac{1}{2}\frac{\partial H(\vec{u}_t|S)}{\partial \mu_{i,j,k,n}(t)}dt + dW_{i,j,k,n}(t) \tag{10}$$

for $n = 1, 2, 3$ where $W_{i,j,k,n}(t)$ is a vector-valued Wiener process with independent components and $H(\vec{u}|S)$ is the Gibbs' potential of the posterior and is given by the sum of the distance and elasticity potentials

$$H(\vec{u}|S) = H(S|\vec{u}) + H(\vec{u}) \quad .$$

From Eq(5) and Eq(6) we can write this as

$$\begin{aligned} H(\vec{u}|S) \;&=\; \frac{1}{2\sigma^2}\int_\Omega \sum_{n=1}^{N} |T_n(\vec{x} - \vec{u}(\vec{x})) - S_n(\vec{x})|^2 d\vec{x} \\ &+\frac{\alpha}{2}\sum_{i=1}^{3}\sum_{j=1}^{3}\int_\Omega \lambda_0 \left(\frac{\partial u_i(\vec{x})}{\partial x_i}\right)\left(\frac{\partial u_j(\vec{x})}{\partial x_j}\right) + \frac{\mu_0}{2}\left(\frac{\partial u_i(\vec{x})}{\partial x_j} + \frac{\partial u_j(\vec{x})}{\partial x_i}\right)^2 d\vec{x} \end{aligned} \tag{11}$$

Under proper conditions [7] the sample mean of the parameter estimates generated from this stochastic gradient algorithm converges to the conditional mean of the posterior, which becomes the estimated transformation field.

3.3 Concatenation of Transformations

To construct the final transformation, the three transformations described previously are combined in a coarse-to-fine approach. At first we find the global scale, translation and rotation displacement vector carrying the template to the target. Eq(1) gives

$$\vec{x} \to \vec{x}' = \quad (A\vec{x} + b) \tag{12}$$

Taking Eq(12) as the initial condition we then find the manifold transformations and the elastic basis transformations given by Eq(3)and Eq(7) to obtain our final transformation as

$$\vec{x}' \to \vec{x}' - \sum_{i=1}^{N} \beta_i K(\vec{x}', \vec{x}_i') + \sum_{i,j,k=0}^{d}\sum_{n=1}^{3} \mu_{n,i,j,k}\vec{\phi}_{n,i,j,k}(\vec{x}') \tag{13}$$

4 RESULTS

Our experiment was aimed to remove the anatomical variation from functional modalities such as PET. This is done by finding the transformations that map the anatomical modality of the atlas to the coordinate system of the study . This transformation is then used to carry the functional modality of the coordinate system of the atlas to that of the study.

The MRI and PET data were collected in the Department of Radiology at Washington University School of Medicine. The MRI data were collected on a 1.5T Siemens Magnetom Vision system using the MPRAGE protocol. This resulted in high-resolution MRI data sets consisting of $256 \times 256 \times 128$ voxels with voxel dimension $1 \times 1 \times 1.25$ mm^3. The PET data were collected on a 954B Siemens ECAT system and was reconstructed to a $128 \times 128 \times 31$ voxel volume.

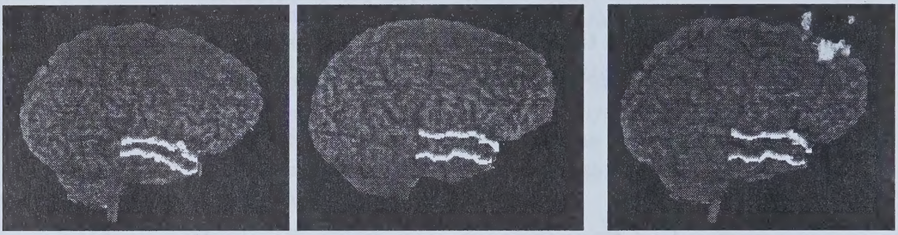


Figure 1: left panel shows the template MRI, middle panel the target and right panel the template mapped to the target. Shown superimposed are the sulcal lines associated with the right Supertemporal sulcus and the right Sylvian sulcus.

We have done the experiments on the entire 3D raw brain image of two patients. The numerical calculations were performed on a 128×128 DECmpp 2200Sx (MasPar), Massively parallel, single-instruction multiple-data (SIMD) computer. Both qualitative and quantitative registration accuracy is presented after the manifold transformation and after the concatenated manifold and elastic basis transformation.

The first step was for an anatomist to insert some well defined sulcal lines on the template and subject brains. The global scale, translation and rotation, as well as, the manifold transformation was based only on these sulcal lines. To illustrate this starting point of the method, Fig. 1 shows the whole brain surface rendered with the superimposed sulcal lines. Left panel shows the template, middle panel the target and right panel the result of the transformation based only on the sulcal lines. Notice the proper superimposition of the target sulcus on the deformed brain.

The low-dimensional sulcal-line transformation was concatenated with the higher-dimensional elasticity transformation. Initially $d = 1$ in Equation(7) and every 60 iterations d was incremented by one. We have found that after 300 iterations the transformation does not change appreciably for the noise level selected ($\sigma = 0.01$). Therefore the basis transformation was estimated by running the SDE for 300 iterations.

Shown in Fig.2 are the output images after each stage of transformation. (a) shows a slice of the MRI images for the template, (b) shows the template mapped to target through manifold transformation, (c) shows the template mapped to the target through the global basis transformation pre-conditioned by the manifold transformation and (d) shows the target MRI. The bottom row (e-g) shows the corresponding difference images from the target. By looking at the difference images we can find the improvement of the mapping when the higher-dimensional basis transformation is pre-conditioned by the lower-dimensional manifold transformation.

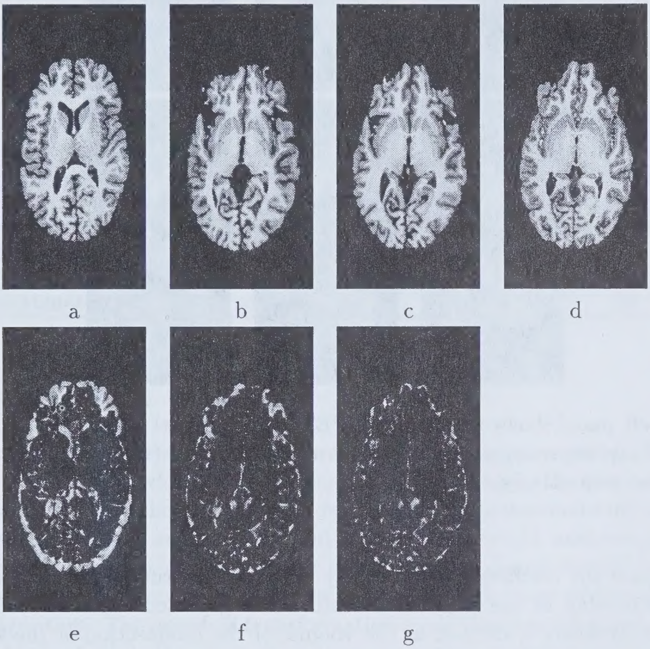


Figure 2: Slice 150 from the human MRI brain: (a) MRI data of the template, (b) deformed brain through manifold transformation, (c) deformed brain through basis transformation with pre-conditioned by manifold transformation and (d) the target. The bottom row (e-g) shows the corresponding difference images from the target for the slice above.

The final part of our experiment was to apply the transformation obtained by mapping the MRI image to the PET image. The subjects' PET images were registered to the subjects' MRI images with the Wood's method [5] Then the registered PET image of the template was carried via the transformation to that of the target. Fig. 3 shows Slice 156 and Slice 168 respectively of the PET image associated with the template (left panel), the PET associated with the target (middle panel) and the template PET image mapped to target through the higher-dimensional elastic basis transformation with pre-conditioning by the manifold transformation(right panel). Notice how the initial template PET varied from the target PET and how close the deformed PET is to the target PET image.

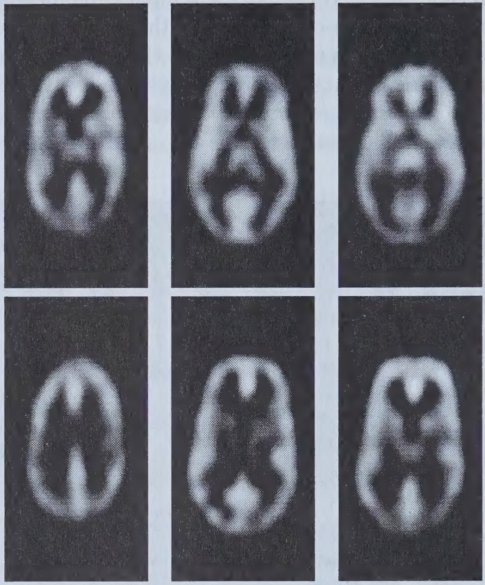


Figure 3: Left panel shows the template PET, middle panel the target PET and right panel the template mapped to the target of section 156 (top row) and section 168 (bottom row) respectively.

To compare the methods quantitatively we have constructed Table 1 which shows the relative overlap of the segmented brain and segmented ventricles. The relative overlap of a structure is defined as the volume of the intersection of the structure in both segmentations divided by the volume of their union.

Notice initially before transformation the relative overlap of both the segmented brain and the ventricles is very low. After the manifold transformation both the overlaps increased. While the relative overlap of the brain increased from 0.662 to 0.800, that of the ventricles increased from 0.031 to 0.351. When we concatenate the elastic basis transformation with the initial manifold transformation there is a sharp improvement in the overlap as the relative overlap of the brain becomes 0.919 while that of the ventricles becoming 0.526. These results demonstrate that the registration accuracy improved after each stage of our hierarchical transformation algorithm.

Table 2 shows the execution time of each stage of the transformations in a 128 × 128 Maspar. Notice that the execution time of the Manifold transformation is small compared to the execution time of the elastic basis transformation. This increase in computation time is related to the increase in the number of estimated parameters.

Table 1: Relative Overlap of the Brain and Ventricles under Different Transformations

Transformation	**Relative overlap of Brain**	**Relative overlap of Ventricles**
Before Transformation	0.662	0.031
After Manifold transformation	0.800	0.351
After Elastic Basis transformation with pre-conditioned by manifold	0.919	0.526

Table 2: 3D Program Execution Time on 128 × 128 MasPar.

Experiment	**data size**	**# of param. estimated**	**# of itrs**	**total time (min)**
Manifold	128 × 256 × 256	100	-	0.3
Elastic basis	32 × 64 × 64	648	300	18
Concatenated	128 × 256 × 256	748	300	19

5 Conclusions

In this paper we have presented a coarse-to-fine approach to register PET data by finding a transformation which maps the template MRI image to the target MRI image. This approach improves the PET to PET registration by accommodating not only the global scale, translation, and rotation, but also local non-rigid anatomical variation. Qualitative and quantitative results were presented that demonstrated improvement in both global and local registration of the MRI after each stage of the coarse-to-fine matching algorithm. The manifold transformation made the algorithm computationally efficient and robust while the elastic basis transformation made it more accurate. The mapping of PET data among different subjects via this approach paves the way towards finding a highly localized region of any functional activity in population brains.

6 Acknowledgments

This work was supported by the ONR N000149251418, NIH-NCRR-RR01380, and the Craniofacial Imaging laboratory, St. Louis Children's Hospital, Washington University Medical Center.

References

[1] M.I. Miller, G.E. Christensen, Y. Amit, and U. Grenander. Mathematical textbook of deformable neuroanatomies. *Proceedings of the National Academy of Science*, 90(24), December 1993.

[2] G. E. Christensen, R. D. Rabbitt, and M.I. Miller. 3D brain mapping using a deformable neuroanatomy. *Physics in Medicine and Biology*, 39:609–618, 1994.

[3] G. Christensen. *Deformable Shape Models for Anatomy.* Ph.D. Dissertation, Department of Electrical Engineering, Sever Institute of Technology, Washington University, St. Louis, MO, Aug 1994.

[4] P.T. Fox, J.S. Perlmutter, and M.E. Raichle. A stereotactic method of anatomical localization for positron emission tomography. *Journal of Computer Assisted Tomography*, 9, No.1:141–153, 1985.

[5] R.P. Woods, S.R. Cherry, and J.C. Mazziotta. Rapid Automated Algorithm for Aligning and Reslicing PET Images. *Journal of Computer Assisted Tomography*, 16(4):620–633, July/August 1992.

[6] Sarang C. Joshi, Michael I. Miller, Gary E. Christensen, Ayananshu Banerjee, Thomas A. Coogan, and Ulf Grenander. Hierarchical brain mapping via a generalized dirichlet solution for mapping brain manifolds. In *Proc. of the SPIE's 1995 International Symposium on Optical Science, Engineering, and Instrumentation*, San Diego, CA., 9-14 July, 1995.

[7] Y. Amit, U. Grenander, and M. Piccioni. Structural image restoration through deformable templates. *J. American Statistical Association*, 86(414):376–387, June 1991.

Time-Frequency Analysis Using Dynamic Statistical Models With Applications to Speech Analysis

Shan Lu and Peter C. Doerschuk*

September 19, 1995

Abstract

We describe an approach to time-frequency analysis based on dynamic stochastic models for the time and frequency variation of the signal. In this framework, time-frequency analysis is essentially Bayesian nonlinear estimation. We apply these ideas to synthetic and to speech signals.

1 Introduction

In this paper we describe an approach to time-frequency analysis based on statistical dynamical models and the application of this approach to problems in speech analysis. The basic idea is that the measured signal is modeled as a superposition of subsignals and each subsignal is a jointly amplitude- and frequency-modulated signal. Therefore, in continuous time, the measured signal $s(t)$ is represented as

$$s(t) = \sum_{i=1}^{I} a_i(t) \cos\left(2\pi \int_{-\infty}^{t} f_i(\tau)\mathrm{d}\tau\right) \tag{1}$$

where $a_i(t)$ is the ith amplitude message and $f_i(t)$ is the ith frequency message. The goal of the signal processing is to extract $a_i(t)$ and $f_i(t)$ $(i = 1, \ldots, I)$ from the measured signal.

The problem as stated is not well posed because the choice $f_i(t) = 0$ for $i = 1, \ldots, I$, $a_i(t) = 0$ for $i = 2, \ldots, I$, and $a_1(t) = s(t)$ provides an exact representation of $s(t)$. Therefore, we need to introduce additional information about the $a_i(t)$ and $f_i(t)$.

The natural goal of this approach is to represent a broad band signal $s(t)$ in terms of narrow band signals $a_i(t)$ and $f_i(t)$. Even if $s(t)$ does not have a sharply peaked spectrum, this type of representation is promising because the

$$a_i(t) \cos\left(2\pi \int_{-\infty}^{t} f_i(\tau)\mathrm{d}\tau\right)$$

---

[*] The authors are with the School of Electrical and Computer Engineering, Purdue Univ., West Lafayette, IN 47907–1285 USA. Corresponding author: Peter C. Doerschuk, 1285 Electrical Engineering Building, Purdue Univ., West Lafayette, IN 47907–1285, USA, (317) 494–1742, (317) 494–3358 (Fax), doerschu@ecn.purdue.edu (Internet). This work was supported by a Whirlpool Faculty Fellowship, U. S. National Science Foundation grant MIP-9110919, and the School of Electrical and Computer Engineering, Purdue University.

nonlinearity will cause bandwidth expansion exactly as occurs in traditional frequency modulation. One possibility for the additional information would be to assume that $a_i(t)$ and $f_i(t)$ are strictly bandlimited to B_{a_i} and B_{f_i} respectively and then solve the approximation problem of minimizing

$$\int_{-\infty}^{+\infty} (s(t) - \hat{s}(t))^2 \mathrm{d}t$$

where $s(t)$ is the data, $\hat{s}(t)$ is the representation of Eq. 1, and the minimization occurs over bandlimited $a_i(t)$ and $f_i(t)$. However, this approach is difficult because of the cosine and multiplicative nonlinearities. Furthermore, the approach is fundamentally batch rather than recursive in orientation and, especially for the speech application, a recursive approach is necessary.

The approach we pursue is to introduce statistical models for $a_i(t)$ and $f_i(t)$. In addition, corresponding to the approximation problem, we hypothesize that the measured data is $s(t) + v(t)$ where $v(t)$ is a noise with known statistical properties. Then, under appropriate assumptions, the goal of reconstructing $a_i(t)$ and $f_i(t)$ from $s(t) + v(t)$ can be posed as a Bayesian estimation problem and solved using nonlinear filtering ideas. We call this statistical nonlinear filtering approach the *Model-Based Demodulation Algorithm* (MBDA).

2 Notation

Expectation is denoted by "E". If x is a random sequence then $m_x(k) \doteq E[x(k)]$, $R_x(k_1, k_2) \doteq E[x(k_1)x(k_2)]$, and $P_x(k_1, k_2) \doteq E[(x(k_1) - m_x(k_1))(x(k_2) - m_x(k_2))]$. Furthermore, in the case where $R_x(k_1, k_2)$ is a function of only $k_1 - k_2$, the discrete-time Fourier transform of $R_x(0, k)$ is denoted by $S_x(\Omega)$, specifically,

$$S_x(\Omega) = \sum_{k=-\infty}^{+\infty} R_x(0, k) e^{-j\Omega k}.$$

Independent and identically distributed is abbreviated by i.i.d. The Gaussian probability density function (pdf) with mean m and covariance Λ is denoted by $\mathcal{N}(m, \Lambda)$. The notation "$x \sim p$" means that the random variable (RV) x is distributed according to the pdf p. If k_1 and k_2 are time indices, let $k_< \doteq \min(k_1, k_2)$ and $k_> \doteq \max(k_1, k_2)$. The Kronecker delta function is denoted by δ_{k_1,k_2}.

3 The Model and Speech

There has been extensive recent interest in taking a speech signal $s(t)$ and extracting amplitude $a(t)$ and phase $\phi(t)$ modulations, i.e., $y(t) = a(t)\cos(\phi(t))$, using Teager's energy operator [2, 3, 5, 7, 8, 9, 12, 11]. Both the case of a linear superposition of terms [8], i.e., $y(t) = \sum_i a_i(t)\cos(\phi_i(t))$, and a single term observed in the presence of noise [2] have been investigated. In both cases, the signal is first passed through a bank of filters and then the energy operator is applied to the output of each filter. In the case of a superposition of terms, the bandwidth of the ith filter is determined by the bandwidth of the term $a_i(t)\cos(\phi_i(t))$ and the outputs of the ith energy operator are $a_i(t)$ and $\phi_i(t)$. Therefore, each filter is responsible for a particular term. In the case of a single term in the presence of noise, the bandwidths of the filters are determined by the trade-off between suppressing the noise and passing as much signal energy as possible and the single signal is tracked (by an energy measure) as it moves from filter to filter.

As described in Section 1, in the MBDA approach we simultaneously consider a linear superposition of terms and the presence of noise. In a qualitative sense, the nonlinear filter acts as a bank of bandpass filters where the center frequency of the ith filter tracks the instantaneous frequency of the $a_i(t)\cos(\phi_i(t))$ term and the bandwidth of the ith filter is set to achieve the optimal trade-off between passing signal energy and rejecting noise based on the statistical model. In this point of view, the parameters of the energy operator approach, specifically the bandwidth and center frequencies of the Gabor filters, are seen to qualitatively correspond to the parameters in the statistical model of MBDA.

In speech, a discrete-time version of the model of Section 1 is more natural and the frequency f_i is split into two parts: a slowly-varying center frequency denoted by $f_i(k)$ (the formant frequency with variation on time-scales greater than the pitch period) and a rapidly-varying message frequency denoted by $\nu_i(k)$ (the instantaneous frequency with variation on time-scales shorter than the pitch period). In addition, for each term in the linear superposition of subsignals, there is an instantaneous amplitude signal, denoted by $a_i(k)$, and a total phase signal, denoted by $\phi_i(k)$. If detailed statistical knowledge concerning $a_i(k)$, $f_i(k)$, and $\nu_i(k)$ is available, then it can be incorporated into the mathematical model. However, in the speech application, only rather imprecise information about power and bandwidth is available. Therefore, we have chosen simple dynamics: The instantaneous amplitude and instantaneous frequency signals a_i and ν_i are modeled as first-order autoregressive (AR) processes which allows independent control of the power and the bandwidth. The formant frequency f_i is modeled as a random walk. This choice was made because we expect the formant frequency to remain constant over periods of milliseconds in duration and a random walk is the only Gauss-Markov model in which such behavior has a large probability of occurring. The dynamics of the total phase signal $\phi(k)$ are completely determined by its definition: $\phi_i(k) = \phi_i(0) + 2\pi T \sum_{l=0}^{k-1}(f_i(l) + \nu_i(l))$ where T is the sampling interval. The measured signal, denoted by $y(k)$, is the linear superposition of the contribution from each formant, specifically, $a_i(k)\cos(\phi_i(k))$, plus additive measurement noise. The complete model is therefore

$$\begin{aligned}
a_i(k+1) &= \alpha_{a_i} a_i(k) + q_{a_i} w_{a_i}(k) && (2)\\
\nu_i(k+1) &= \alpha_{\nu_i} \nu_i(k) + q_{\nu_i} w_{\nu_i}(k) && (3)\\
f_i(k+1) &= f_i(k) + q_{f_i} w_{f_i}(k) && (4)\\
\phi_i(k+1) &= \phi_i(k) + 2\pi T f_i(k) + 2\pi T \nu_i(k) && (5)\\
y(k) &= \sum_{i=1}^{I} a_i(k)\cos(\phi_i(k)) + r v(k) && (6)
\end{aligned}$$

where the process noises w_{a_i}, w_{ν_i}, and w_{f_i} and the observation noise v are all iid $N(0,1)$ sequences; the initial conditions are $a_i(0) \sim N(0, q_{a_i}^2/(1-\alpha_{a_i}^2))$, $\nu_i(0) \sim N(0, q_{\nu_i}^2/(1-\alpha_{\nu_i}^2))$, $f_i(0) \sim N(m_{f_i,0}, p_{f_i,0}^2)$, and $\phi_i(0) \sim N(0, p_{\phi_i,0}^2)$; and the process noises, observation noise, and initial conditions are all independent. Notice that the initial conditions require that $|\alpha_{a_i}| < 1$ and $|\alpha_{\nu_i}| < 1$ (since otherwise the stated variances are negative) in which case a_i and ν_i are wide sense stationary random sequences. For later convenience, define $\theta = (\alpha_{a_i}, q_{a_i}, \alpha_{\nu_i}, q_{\nu_i}, q_{f_i}, r, m_{f_i,0}, p_{f_i,0}, p_{\phi_i,0})$. We estimate the parameter vector θ by matching the second order statistics of the model to training data.

4 Bandwidth Expansion

Nonlinear systems of the type described in Section 3 exhibit bandwidth expansion analogous to the bandwidth expansion of a frequency-modulated communication system. The following result describes the bandwidth expansion in terms of the autocorrelation function.

Theorem 1 *Let* $a(k)$ *and* $\phi(k)$ *be random sequences where* ϕ *is Gaussian. Let* ψ *be a RV that is distributed uniformly on* $[-\pi, \pi]$. *Let* a, ϕ, *and* ψ *be independent. Define* $y(k) = a(k)\cos(\phi(k) + \psi)$. *It follows that* $m_y(k) = 0$ *and*

$$R_y(k_1, k_2) = P_y(k_1, k_2) = R_a(k_1, k_2)\frac{\cos(m_\phi(k_1) - m_\phi(k_2))}{2}\exp\left(-\frac{1}{2}Q_\phi(k_1, k_2)\right)$$

where $Q_\phi(k_1, k_2) \doteq P_\phi(k_1, k_1) - 2P_\phi(k_1, k_2) + P_\phi(k_2, k_2)$. *More generally, if* $y(k) = \sum_i y_i(k) + rv(k)$ *where* $y_i(k) = a_i(k)\cos(\phi_i(k) + \psi_i)$; a_i, ϕ_i, ψ_i *and* v *are independent; and, for each* i, *the quantities* a_i, ϕ_i, *and* ψ_i *are as above, then it follows that* $m_y(k) = rm_v(k)$ *and* $P_y(k_1, k_2) = \sum_i P_{y_i}(k_1, k_2) + r^2 P_v(k_1, k_2)$ *where* P_{y_i} *is given by Eq. 1.*

Proof: The proof is by direct computation using standard results on the characteristic function of a Gaussian random variable in order to deal with the cosine nonlinearity. □

Since $Q_\phi(k_1, k_2)$ can alternatively be expressed as

$$Q_\phi(k_1, k_2) = E[\{[\phi(k_1) - m_\phi(k_1)] - [\phi(k_2) - m_\phi(k_2)]\}^2],$$

it follows that $Q_\phi(k_1, k_2) \geq 0$.

For the system of Eqs. 2–6 with $|\alpha_{a_i}| < 1$ and $|\alpha_{\nu_i}| < 1$, it follows that

$$\begin{aligned}
R_{a_i}(k_1, k_2) &= (q_{a_i}^2/(1 - \alpha_{a_i}^2))\alpha_{a_i}^{|k_2 - k_1|}, \\
m_{\phi_i}(k) &= 2\pi T m_{f_i,0} k, \\
R_v(k_1, k_2) &= \delta_{k_1, k_2}, \\
Q_{\phi_i}(k_1, k_2) &= (2\pi T)^2 \Big\{ p_{f_i,0}^2 (k_2 - k_1)^2 \\
&\quad + \frac{q_{f_i}^2 |k_2 - k_1| (k_>(k_> - 1) - k_<(k_< - 1))}{2} \\
&\quad - \frac{q_{f_i}^2 (|k_2 - k_1| + 1)|k_2 - k_1|(|k_2 - k_1| - 1)}{6} \\
&\quad + \frac{q_{\nu_i}^2}{1 - \alpha_{\nu_i}^2}\left[\frac{(1 + \alpha_{\nu_i})|k_2 - k_1|}{1 - \alpha_{\nu_i}} - \frac{2\alpha_{\nu_i}(1 - \alpha_{\nu_i}^{|k_2 - k_1|})}{(1 - \alpha_{\nu_i})^2}\right]\Big\}.
\end{aligned}$$

Using these results in Eq. 1 provides the necessary $R_y(k_1, k_2)$ for the system of Eqs. 2–6.

Because of the second term in the braces, $Q_{\phi_i}(k_1, k_2)$ is not a function of $k_1 - k_2$ and therefore $R_y(k_1, k_2)$ is not wide-sense stationary. This reflects the fact that f_i is not wide-sense stationary. In order to easily demonstrate bandwidth expansion, it is convenient to have a wide-sense stationary R_y and therefore we set $q_{f_i} = 0$. Unfortunately, we are unable to analytically compute $S_y(\Omega)$. Therefore, we give an numerical example using one subsignal and evaluating $S_y(\Omega)$ numerically after truncating the infinite sum to $\sum_{k=-4096}^{+4096}$. The parameters of the example are $\alpha_{a_1} = .99$, $q_{a_1} = 1$, $\alpha_{\nu_1} = .99$, $q_{\nu_1} = 20$, $q_{f_1} = 0$, $r = 0$, $m_{f_1,0} = 1000$ Hz, $T = 1/16000$ s, $p_{f_1,0} = 0$, and $p_{\phi_1,0} = 0$. In Figure 1 we show the power spectral densities for $a_1(k)$ and $\nu_1(k)$, which are identical except for a constant scaling by 20^2, and the power spectral density for $y(k)$. Depending on the details of the definition of bandwidth, a bandwidth expansion of roughly 3 times has taken place in this example.

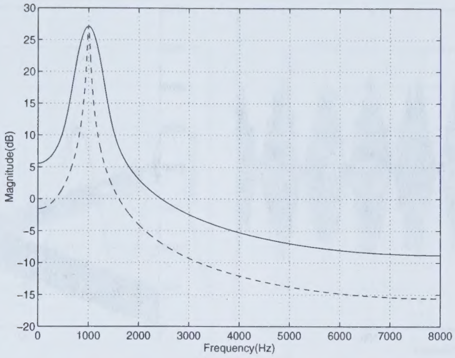


Figure 1: Bandwidth Expansion: solid curve is $S_y(\Omega)$; dashed curve is the power spectral density of $a_1(k)$ shifted in frequency to $m_{f_1,0}$ and scaled in amplitude to match $S_y(\Omega)$, i.e., $(S_y(m_{f_1,0})/S_{a_1'}(m_{f_1,0}))S_{a_1'}(\Omega)$ where $a_1'(k) = a_1(k)\cos(2\pi m_{f_1,0}kT)$. $S_{\nu_1}(\Omega)$ is proportional to $S_{a_1}(\Omega)$.

5 Nonlinear Filters

If $a_i(k)$ was constant then Eqs. 2–6 describe a frequency modulated communication system, the Extended Kalman Filter (EKF) [1, Section 8.2] is essentially a phase-locked loop (PLL), and the PLL is an excellent estimator. Therefore, we compute the estimates $\hat{a}_i(k|k)$, $\hat{\nu}_i(k|k)$, $\hat{f}_i(k|k)$, and $\hat{\phi}_i(k|k)$ (hereafter, we will not indicate the conditioning which is always $k|k$) by using the EKF for this more complicated model. The computational requirements are minimal: the state equation is already linear, the one-step state transition matrix (denoted by F) is block diagonal (1 block per formant) and each block is sparse so multiplication by F is inexpensive, and the observation is a scalar so the one matrix inversion is actually division by a scalar. The result of the EKF are the estimates $\hat{a}_i(k)$, $\hat{\nu}_i(k)$, $\hat{f}_i(k)$, and $\hat{\phi}_i(k)$. From these estimates we can compute a reconstructed speech signal, denoted by $\hat{y}(k)$, by $\hat{y}(k) = \sum_i \hat{a}_i(k)\cos(\hat{\phi}_i(k))$.

More sophisticated nonlinear filters than the EKF could also be used, e.g., Refs. [10, 1, 6]. However, we have achieved interesting results using the simple EKF and, when using more sophisticated filters, problems of robustness and failure of the model statistics to match the data statistics become more severe.

6 Synthetic Example

In this section we describe a double-chirp synthetic example which is patterned after the single-chirp example of Ref. [8, Fig. 2]. The signal is

$$y(k) = \cos(2\pi f_a kT)\cos(2\pi(f_{m1} + f_c kT)kT) + 0.2\cos(2\pi f_a kT)\cos(2\pi(f_{m2} - f_c kT)kT)$$

where $T = 1/16000$ s, $f_a = 30$ Hz, $f_{m1} = 200$ Hz, $f_{m2} = 2000$ Hz, and $f_c = 6000$ Hz/s. The EKF has parameters $\alpha_{a_1} = \alpha_{a_2} = .99$, $\alpha_{\nu_1} = \alpha_{\nu_2} = .99$, $q_{a_1} = \sqrt{.001}$, $q_{a_2} = \sqrt{.001 \times .04}$, $q_{f_1} = q_{f_2} = \sqrt{12}$, $q_{\nu_1} = q_{\nu_2} = \sqrt{.01}$, $r = 1.0\times 10^{-7}$, $m_{f_1,0} = f_{m1}$, $m_{f_2,0} = f_{m2}$,

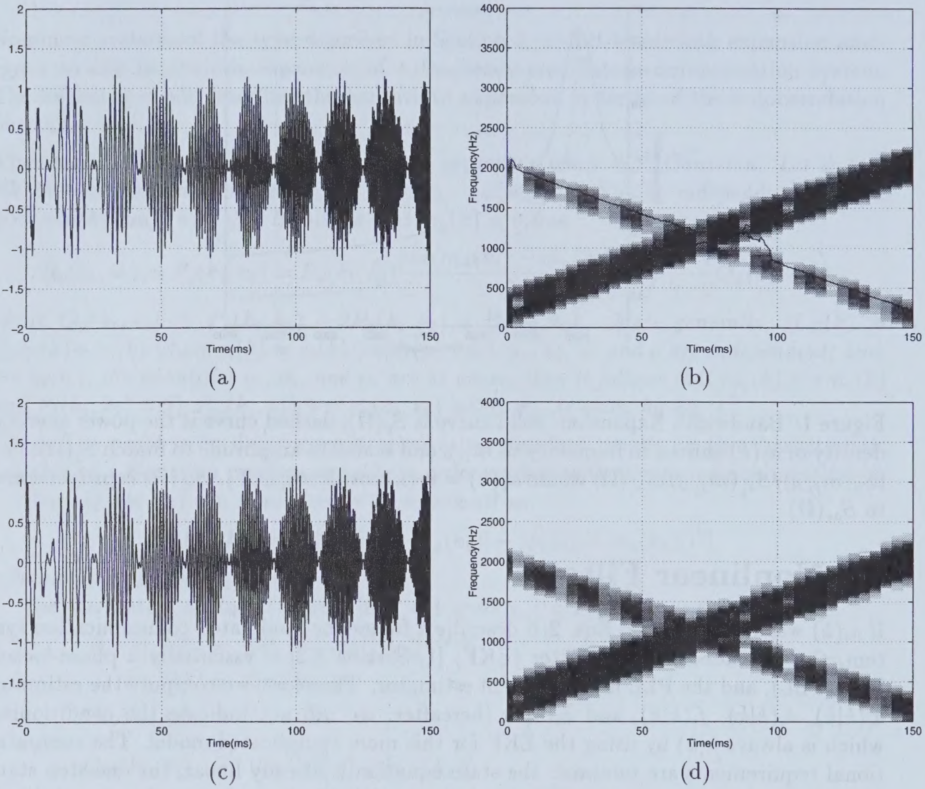
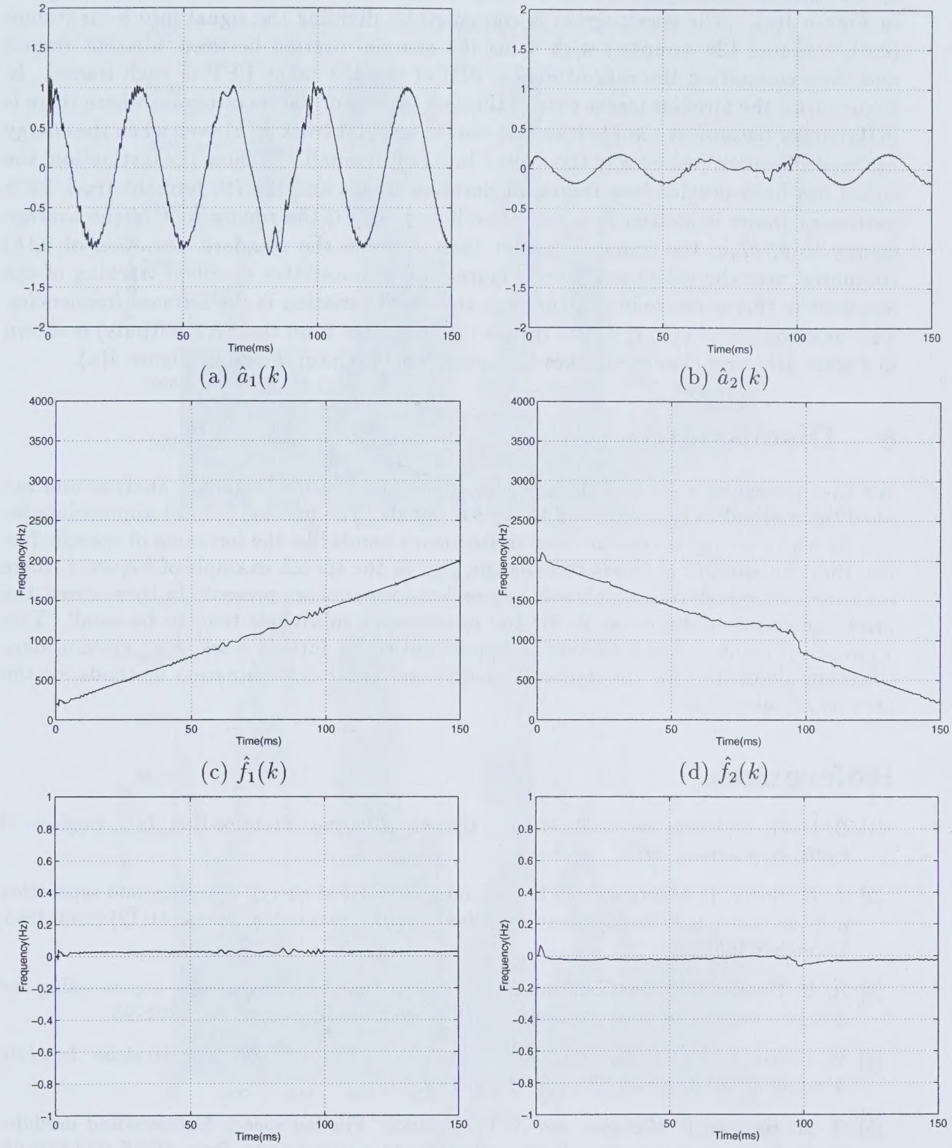


Figure 2: Two-chirp Example: (a) Original signal, (b) original spectrogram and estimated formant tracks, (c) reconstructed signal, (d) reconstructed spectrogram. Both spectrograms are computed using an 8 ms window length with 4 ms overlap and a 512-point FFT.

and all the other initial condition means and all the initial condition variances equal to zero. Ideal performance of the EKF in this example would lead to $\hat{a}_1(k) = \cos(2\pi f_a kT)$, $\hat{f}_1(k) = f_{m1} + 2f_c kT$, $\hat{a}_2(k) = 0.2\cos(2\pi f_a kT)$, $\hat{f}_2(k) = f_{m2} - 2f_c kT$, and $\hat{\nu}_1(k) = \hat{\nu}_2(k) = 0$. The actual results, shown in Figures 2 and 3 are quite good.

7 Speech Example

In this section we describe the application of these ideas to the sentence "A roll of wire lay near the wall." from the TIMIT database [4, dr3/mdbb1/sx16]. The model has 4 formants with initial conditions $m_{f_i,0}$ of 560, 1060, 2400, and 3050 Hz for $i = 1$, 2, 3, and 4 respectively. For all 4 formants, $\alpha_{a_i} = \alpha_{\nu_i} = .99$, $q_{\nu_i} = 12$, $p_{f_i,0} = 0$, and $p_{\phi_i,0} = 0$. The values of q_{a_i} and q_{f_i} vary from formant to formant: $q_{a_i} = 50, 30, 15, 1$;



(a) $\hat{a}_1(k)$ (b) $\hat{a}_2(k)$

(c) $\hat{f}_1(k)$ (d) $\hat{f}_2(k)$

(e) $\hat{\nu}_1(k)$ (f) $\hat{\nu}_2(k)$

Figure 3: Two-chirp Example. Note the small vertical scales for the $\hat{\nu}_1(k)$ and $\hat{\nu}_2(k)$ plots.

and $q_{f_i} = \sqrt{.5}, \sqrt{20}, \sqrt{.5}, \sqrt{.5}$ for $i = 1, 2, 3, 4$ respectively. Finally, $r = \sqrt{1/12}$. The spectrogram of the original speech with superimposed plots of the estimates $\hat{f}_i(k)$ is shown in Figure 4(a). [The spectrogram is computed by dividing the signal into 8 ms frames (each contains 128 samples) with 4 ms (64 sample) overlap between adjacent frames and then computing the magnitude (in dB) of the 128 point FFT of each frame]. In Figure 4(a), the formant tracks extend through regions of the spectrogram where there is little energy because at sample k we plot the ith formant track $\hat{f}_i(k)$ even when the energy in the ith formant (essentially the energy in $a_i(k)$) is small. To show the distinction, the signal has been divided into frames of duration 20 ms and the ith formant track for a particular frame is plotted as a solid line if and only if the square root of the average energy in $\hat{a}_i(k)$ in the frame is greater than .3 times the standard deviation of $\hat{a}_i(k)$ computed over the entire sentence. Figure 4(a) demonstrates excellent tracking of the formants in this sentence in spite of large and rapid variation in the formant frequencies. The spectrogram of $\hat{y}(k)$ (i.e., the speech reconstructed from the EKF outputs) is shown in Figure 4(b) and closely matches the speech spectrogram shown in Figure 4(a).

8 Discussion

We have presented a statistical model-based method of time-frequency analysis and applied the method to synthetic and to speech signals. The method is most appropriate for signals where energy is concentrated in frequency bands like the formants of speech. Notice that the number of bands can change, e.g., in the speech example of Figure 4, there are time intervals during which only one or two formants are present. In these cases, the instantaneous amplitude signals for the unnecessary subsignals tend to be small. This approach presents a large number of opportunities for further work, e.g., system identification algorithms for the dynamic models and order determination methods for the number of submodels.

References

[1] B. D. O. Anderson and J. B. Moore. *Optimal Filtering.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1979.

[2] A. C. Bovik, P. Maragos, and T. F. Quatieri. AM-FM energy detection and separation in noise using multiband energy operators. *IEEE Trans. Sig. Proc.*, 41(12):3245–3265, December 1993.

[3] R. B. Dunn, T. F. Quatieri, and J. F. Kaiser. Detection of transient signals using the energy operator. In *Proc. IEEE ICASSP-93*, volume III, pages 145–148, 1993.

[4] W. Fisher, V. Zue, J. Bernstein, and D. Pallett. An accoustic-phonetic database. In *113th Meeting of the Acoustical Society of America*, 1987.

[5] H. M. Hanson, P. Maragos, and A. Potamianos. Finding speech formants and modulations via energy separation: With application to a vocoder. In *Proc. IEEE ICASSP-93*, volume II, pages 716–719, 1993.

[6] A. H. Jazwinski. *Stochastic Processes and Filtering Theory.* Academic Press, New York, 1970.

[7] J. F. Kaiser. Some useful properties of Teager's energy operators. In *Proc. IEEE ICASSP-93*, volume III, pages 149–152, 1993.

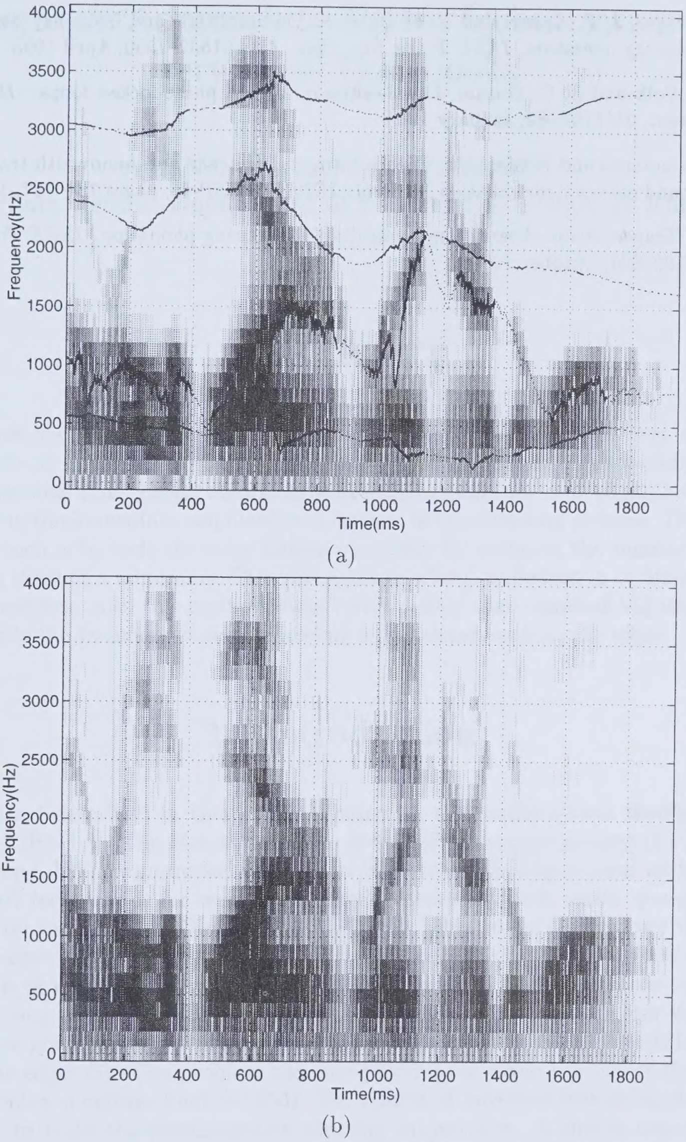


Figure 4: The sentence "A roll of wire lay near the wall." (a) Original spectrogram and estimated formant tracks. (b) Reconstructed spectrogram.

[8] P. Maragos, J. F. Kaiser, and T. F. Quatieri. Energy separation in signal modulations with application to speech analysis. *IEEE Trans. Sig. Proc.*, 41(10):3024–3051, October 1993.

[9] P. Maragos, J. F. Kaiser, and T. F. Quatieri. On amplitude and frequency demodulation using energy operators. *IEEE Trans. Sig. Proc.*, 41(4):1532–1550, April 1993.

[10] D. R. Polk and S. C. Gupta. Quasi-optimum digital phase-locked loops. *IEEE Trans. Commun.*, 21(1):75–82, January 1973.

[11] A. Potamianos and P. Maragos. Speech formant frequency and bandwidth tracking using multiband energy demodulation. In *Proc. IEEE ICASSP-95*, pages 784–787, 1995.

[12] H. M. Teager. Some observations on oral air flow during phonation. *IEEE Trans. ASSP*, 28(5):599–601, October 1980.

An Iterative Approach to Vector Quantization with very small Codebooks

Ajai Narayan and John F. Doherty
Durham Center, Department of Electrical & Computer Engg.
Ames, IA 50011

ABSTRACT

This paper proposes an implementation of a Vector Quantizer with very small codebooks (i.e, 30-32 codevectors) for compressing grayscale images. The technique uses a convex projections (CP) based algorithm for iterative restoration of edges, based on the shade pixels in the immediate neighborhood, as part of the decoding process. The objective of this approach is to code the edge blocks vestigially by reducing the number of vectors representing them to a minimum. This will result in a large reduction in codebook size and hence fast searches. Also this method should work better than classical VQ techniques on images outside the training set since encoding is less dependent on the edges.

1 Introduction

The method described in this paper is based on the mathematical theory of convex projections (CP) [1,2]. The idea of using the theory of convex projections (CP) for recovering DCT and JPEG-compressed images has been researched by several authors [3,4,5]. The proposed technique is for recovering images compressed with vector quantizers (VQ) at very low bit rates. Vector Quantization (VQ) has been shown to be a very viable alternative technique to the transform (DCT) domain technique [6]. VQ does not suffer from the problems of ringing like transform coders. However, it suffers from one key problem that occurs very consistently. Any image encoded using a vector quantizer with a finite codebook has good reproducibility of shade regions but suffers from perceptible blocking effects at the edges [7]. The problem becomes pronounced when the size of the codebook reduces to below a certain limit (≈ 256). Some form of localized smoothing at the edges is called for to make the reconstruction pleasing subjectively. A simple smoothing filter applied to the entire decoded image degrades the shade regions while smoothing the edges. This paper proposes a technique that includes a projection based image restoration algorithm as part of the decoding process. The method can be considered as a constrained smoothing of edges with the 'staircase effect' based on the surrounding shade regions. It however does not cause any degradation to the shade areas in the decoded image. The result is a reconstruction that is very pleasing subjectively.

2 Theory

An image of dimensions $N \times N$ pixels can be considered to be a mosaic of constituent image blocks of size $n \times n$ $(n < N)$ pixels. These blocks can in turn be mapped onto a subset of vectors from a n^2-D Hilbert space. If b is a image (dimension: $N \times N$) composed of ordered smaller image blocks (dimension $n \times n$) $f^{(a)}, a = 1, 2, 3, ...\infty$, then the image can mathematically be represented as:

$$b = \sum_{i=0}^{N/n} \sum_{j=0}^{N/n} f^{(a)}(x - ni, y - nj) \tag{1}$$

Each individual block is characterized as either a shade or an edge block depending on the activity level of each block. A set of criteria to quantify the activity level of an image block used in this paper is described in detail in [6].

2.1 Sets and Projections Associated with the Image Data

Let $b \longleftrightarrow B$ $(i.e., B = \mathfrak{F}\{b\})$ denote a 2-D Fourier transform pair. The sets are defined as follows.

(i) $\zeta_1 = \{g_m \mid g_m \in V_1, m = 0, 1, ...M_1\}$ denotes the subset of codevectors g_m in a vector quantizer (VQ) that are obtained from the shade-vector space V_1 of the training images. The association of a vector into the shade vector space is described by Ramamurti and Gersho [6].

(ii) $\zeta_2 = \{h_m \mid h_m \in V_2, m = 0, 1, ...M_2\}$ denotes the subset of codevectors h_m in a vector quantizer (VQ) that are obtained from the edge-vector space V_2 of the training images. The association of a vector into the edge vector space is again described by Ramamurti and Gersho [6].

(iii) $\zeta_3 = \{b \mid B[u, v] = 0, \forall [u, v] \notin \chi_c\}$ denotes the subset of images whose Fourier transform components equal 0 outside a certain region χ_c in the $u - v$ plane.

(iv) $\zeta_4 = \{b \mid \alpha \leq e_n \leq \beta, \forall e_n \in b\}$ denotes a subset of all images whose individual pixels lie between a prescribed closed interval $[\alpha, \beta]$.

(v) $\zeta_{NL} = \{b \mid \forall f \in V_1,\ f = K_m,\ K_m \in \zeta_1,\ m = 0, 1, 2, ...M_1\}$ denotes a subset of all images that have identical shade vectors but different edge vectors.

The projections on these subsets are defined as given below:

i. *Projection onto* ζ_1: If f is any arbitrary vector in the n^2-D Hilbert space then the VQ operation is defined as:

$$\widehat{f} = P_1 f = \begin{cases} \min_m \|f - g_m\|_2 & \text{if } f \in V_1 \\ 0, & \text{if } f \notin V_1 \end{cases} \tag{2}$$

where $\widehat{f}$ is the vector that has the smallest l_2 norm with f in the subset ζ_1.

ii. *Projection onto* ζ_2: If f is any arbitrary vector in the n^2-D Hilbert space then the VQ operation is defined as:

$$\hat{f} = P_2 f = \begin{cases} \min_m \|f - h_m\|_2 & \text{if } f \in V_2 \\ 0, & \text{if } f \notin V_2 \end{cases} \tag{3}$$

where h_m is the vector that has the smallest l_2 norm with f in the subset ζ_2.

iii. *Projection onto* ζ_3: The projection is defined by a lowpass mask in the Fourier domain. If b is any vector obtained as a spatial union of vectors from the n^2-D Hilbert space then the operation in this subset is defined as:

$$P_3\, b = \begin{cases} B[u,v], & \text{if } (u,v) \in \chi_c \\ 0, & \text{if } (u,v) \notin \chi_c \end{cases} \tag{4}$$

where $b \longleftrightarrow B$ is the 2-D Fourier transform pair and b itself is a spatial union of vectors (i.e., f's) from the n^2-D space as represented by (1).

iv. *Projection onto* ζ_4 : If b is any arbitrary vector (image) in the N^2-D Hilbert space and e_1, e_2, e_3... are its elements then the projection operation $\overline{f}$ is defined as:

$$\overline{f} = P_4 b = \begin{cases} \alpha & \text{if } e_n < \alpha \\ e_n & \text{if } \alpha \le e_n \le \beta, \ \forall n \\ \beta & \text{if } e_n > \beta \end{cases} \tag{5}$$

v. *Projection onto* ζ_{NL} : If b is any arbitrary image, with its vectors classified to belong to either V_1 or V_2, then the projection operation P_{NL} is defined as

$$P_{NL}\hat{b} = \sum_{i=0}^{N/n} \sum_{j=0}^{N/n} \left[K_m^{(a)}(x - ni, y - nj)\right] \tag{6}$$

where $K_m^{(a)}$ is the codevector that is identical in all the images at that particular position in the subset ζ_{NL}.

During the operation of VQ, subsets ζ_1 and ζ_2 are merged to form a single codebook for encoding. The operation of VQ inherently introduces additional high frequency components into the decoded image in the form of discontinuities at every block boundary. ζ_3 is defined to eliminate these discontinuities. However, all these subsets are not strictly convex, particularly $\zeta_1 \cup \zeta_2$, due to the inherent non convexity of quantization and digitization operations. This can also be shown by the fact that the sets defined above do not satisfy the fundamental definition of convex sets:

DEFINITION 1. *A subset* ζ *of a Hilbert space is said to be convex if together with any* f_1 *and* f_2 *it also contains* $\mu f_1 + (1-\mu) f_2$ *for all* μ , $0 \le \mu \le 1$.

This is the reason for imperfect restoration by the method of CP as applied to VQ. However, this does not negate the effectiveness of such methods but only serves to define their limits. Algorithms based on the theory of CP can and do work to restore images, and hence improve the reconstruction (from a subjective point of view).

Mathematically, the operation of VQ after classification into shade and edge vectors can be expressed as:

$$\begin{array}{ll} \widehat{b} = (P_s + P_e)\, b = \sum_{i=0}^{N/n} \sum_{j=0}^{N/n} \left[P_1 f^{(a)}(x-ni, y-nj) + P_2 f^{(a)}(x-ni, y-nj)\right] & \\ \text{where} \qquad P_s\, b = \sum_{i=0}^{N/n} \sum_{j=0}^{N/n} P_1 f^{(a)}(x-ni, y-nj) & (7) \\ \text{and} \qquad P_e\, b = \sum_{i=0}^{N/n} \sum_{j=0}^{N/n} P_2 f^{(a)}(x-ni, y-nj) & \end{array}$$

where $\widehat{b}$ is the decoded image composed of ordered image blocks which are projections of the vectors in either ζ_1 or ζ_2 at the corresponding positions in the original image. $P_s b$ is the VQ'ed image with all shade blocks reconstructed and edge blocks masked, while $P_e b$ is the VQ'ed image with all edge blocks reconstructed and shade blocks masked.

2.2 The Algorithm

Image restoration at the decoder is performed with an iterative algorithm which amounts to a repeated application of the projections defined in the previous section.

St.1: The VQ'ed image is reconstructed at the decoder with shade and edge codebooks identical to those at the encoder, which is represented as $\widehat{b} = \widehat{b}_0$

St.2: The reconstructed image is then partially smoothed (using a low pass filter) to reduce the 'blocking effect' inherent in the operation of VQ. This projection operation is symbolically equivalent to $P_3\, \widehat{b}$, a projection of the VQ'ed image onto its low frequency subband.

St.3: The resulting image is then tested and corrected for any pixel intensity limit violations, i.e., $P_4 P_3\, \widehat{b}$.

St.4: The smoothed image is reprojected back onto the set $\zeta_1 \cup \zeta_2$, but with a modification in the projection operation. All shade blocks are replaced with their corresponding unsmoothed versions, while the edge blocks are left unreplaced, $P_{NL} P_4 P_3 \widehat{b}$. Thus, the sequence of projection operations depends upon the space (V_1 or V_2), to which the vector in question belongs to.

A mathematically compact notation for one iteration of the algorithm can then be given by

$$\begin{array}{ll} \widehat{b}_{k+1} = P_{NL} P_4 P_3\, \widehat{b}_k & \\ \widehat{b}_0 \;\;\;\; = (P_s + P_e)\, b & (8) \end{array}$$

The numerical implementation of the nonlinear projection operation is realized by defining an indicator function in the spatial domain as follows:

$$\Gamma = \begin{cases} 1 & \text{if } \widehat{f_0}^{(n)} = P_1 f^{(n)} \\ 0 & \text{otherwise} \end{cases} \qquad (9)$$

where $f^{(n)}$ is the nth vector in the original image and $\widehat{f_0}^{(n)}$ is the corresponding decoded vector. The mathematical expression for image restoration is then given as below:

$$\widehat{f_{k+1}} = \Gamma\, \widehat{f_0} + (1-\Gamma)\, \mathfrak{F}^{-1}\left[\widehat{B}_k\,[u,v]\ \ (\chi_c)\right] \tag{10}$$

where $\widehat{f_0}$ is the vector from the VQ'ed image, $\widehat{f_k}$ is the vector from the kth iteration, $\mathfrak{F}^{-1}$ is the inverse Fourier transform operator and $\widehat{B}_k\,[u,v]\,\chi_c$ is the lowpassed Fourier transform of the entire image after k iterations. The following assumptions are inherent in the interpretation of (10). All higher frequency components are localized exclusively to the edge blocks. Shade blocks contain negligible amounts of high frequency components, and so, are unaffected by the restoration algorithm.

3 Implementation

At the Encoder:

The input image is partitoned into subblocks of 4 × 4 pixels. The mean of each block is predicted causally, both at the encoder and the decoder as a function of the nine adjacent pixels as shown in the Figure 1 [8]. If the difference between predicted mean and the actual mean (at the encoder) of a block is greater than a threshold, the actual mean is transmitted as channel information. If not, the predicted mean is used during decoding.

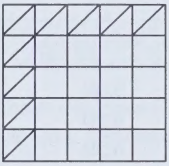


Figure 1: Calculation of the average value

The mean removed blocks are then quantized using a VQ. The codebook for the VQ is composed of two subcodebooks, a shade codebook (20-25 codevectors) and a vestigial edge codebook (8-10 codevectors). This operation is a non-linear approximation of projection of a vector space onto its subset. The indices of the codebook vector nearest to a vector in question is entropy (Huffman/Arithmetic) coded and transmitted through the channel.

At the Decoder:

Indices are received from the channel, and the image is reconstructed using the corresponding codebook vectors. The resulting decoded image has good shade reproduction but annoyingly degraded edges. The CP algorithm is used to restore the degraded edges without altering the shade regions. This restoration is not exact but experiments on real grayscale images produced subjectively pleasing edges with near complete elimination of the staircase effect. The algorithm itself is iterative, alternating between the spatial and the frequency domains.

The result of the projection in Step 2 of the algorithm can be interpreted as follows: The smoothing operation is equivalent to letting the edge subset and the shade subset mutually project onto each other. This operation has very little effect on the shade blocks that are far removed from the edges. However, it drastically affects all the edge blocks and shade blocks in their vicinity. Step 4 effectively nullifies all projections of edge subset onto the shade subset while letting the shade subset project on to the edge subset. Steps 1-4 are performed iteratively to obtain a final image with smallest l_2 norm between the original and the decoded image.

4 Experimental Results

The algorithm was tested on photographic images from the library archives of USC. The codebook was comprised of 24 shade vectors and 6 edge vectors. The well known LBG algorithm was used to generate the codebook. The training set for the codebook generation was obtained from five images. Testing was performed on two images outside the training set 1) Lenna and 2) Peppers and three from within the training set 3) Sailboat and 4) Baboon 5) Couple. Figure 2 summarizes the performance results of the algorithm on 5 photographic images. The l_2 norm of all five images decreased monotonically for 12-15 iterations beyond which it began to increase. The reconstructions of the image 'Lenna' before and after the restoration, at a bit rate of 0.254 bpp is shown in Figure 3 and Figure 4 respectively.

Images	Comp. Ratio (bit/pixel)	SNR
Peppers	0.261	32.43
Lenna	0.254	31.82
Sailboat	0.248	33.67
Babboon	0.267	32.03
Couple	0.307	31.01

Figure 2: The performance results of the algorithm on photographic grayscale images

5 Conclusions

A technique, based on convex projection theory is presented to drastically reduce the codebook size in a VQ. Reconstructions of two images at an approximate compression ratio of 30:1 is shown. The reconstructions are comparable to JPEG compressed reconstructions at the same compression ratio. Small codebooks composed of exclusively shade codevectors increases the joint probabilities of codevector occurrences drastically. Methods like Address VQ [8] which are based on high joint probabilities of codevector occurrences may be combined with CP to obtain much lower compression ratios. Reducing the number of iterations without sacrificing the subjective fidelity of the reconstructions, may make convex projections a viable method to improve the quality of decoded images in real time applications like video coding.

6 REFERENCES

[1] D. C. Youla, "Generalized image restoration by the method of alternating projections," *IEEE Transactions on Circuits and Systems*, vol. CAS-25, October 1978.

[2] H. Stark, "Image Recovery Theory and Application" *Academic Press, New York*, 1987

[3] A. Zakhor, "Iterative procedures for reduction of blocking effects in transform image coding," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 91–95, March 1992.

[4] N. G. Y Yang and A. K. Katsaggelos, "Regularized reconstruction to reduce blocking artifacts of block-dct compressed images," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 421–432, December 1993.

[5] J. K. Su and R. M. Mersereau, "Post processing for artifact reduction in jpeg-compressed images," *Proceedings of ICASSP, Detroit*, pp. 2363–2366, May 1995.

[6] N. M. Nasrabadi, "Use of vector quantizers in image coding," *Proceedings of ICASSP, Tokyo*, pp. 125–127, May 1985.

[7] B. Ramamurti and A. Gersho, "Classified vector quantization of images," *IEEE Transactions on Communications*, vol. COM-34, pp. 1105–1115, November 1986.

[8] N. M. Nasrabadi and Y. Feng, "Image compression using address vector quantization," *IEEE Transactions in Communications*, December 1990.

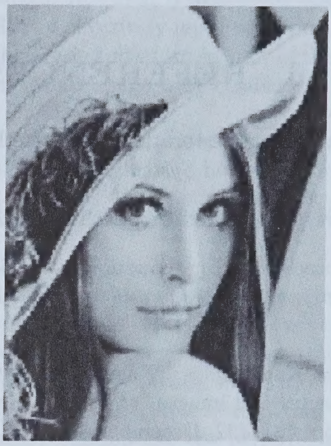


Figure 3: 'Staircase effect' due to vestigial coding of the edge vectors without restoration (0.254 bpp)

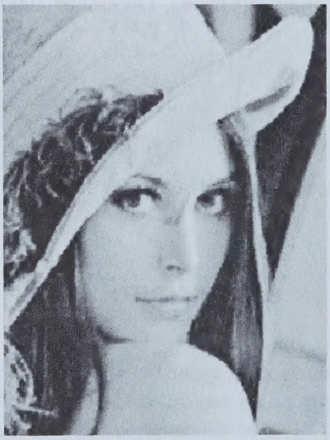


Figure 4: Restoration of the edges by the convex projection algorithm (0.254 bpp)

A layered video coding algorithm for multimedia applications over ATM

Paweł Gburzyński and Gopalkrishnan Shankar

Department of Computing Science
University of Alberta
Edmonton, Alberta
Canada T6G 2H1

http://web.cs.ualberta.ca/~[pawel,shankar]

Abstract

We introduce a new coding algorithm for motion video. Our algorithm, dubbed SPAFLAY, is particularly well suited for multimedia applications (like teleconferencing and distance education) over ATM networks, as it naturally takes advantage of the two cell priority levels in ATM. Based on the dynamically changed focal point of the image, it assigns a higher priority to the cells carrying information about the current area of interest. This spatial layering technique is supplemented by layering in the frequency domain. Even if all low-priority cells are dropped by a congested switch, the quality of the received picture turns out to be acceptable in most cases. The coding method presented here has been used in a distance education package developed at the University of Alberta.

1 Introduction

Owing to its asynchronous mode of operation, ATM naturally lends itself to variable bit rate (VBR) coding for video signals. In contrast to constant bit rate (CBR) coding, VBR schemes tend to utilize better the network bandwidth and generally deliver better picture quality per unit of bandwidth, but they are bursty in nature and more susceptible to congestion. With the CBR approach, each frame is described by the same amount of information and the picture quality varies with the activity of the instantaneous scenes. On the other hand, in VBR coding, data rate may vary drastically, but the picture quality is kept constant (assuming no packet losses). ATM favors VBR coding [4, 10] because it can freely multiplex cells coming from several data streams regardless of their bit-rate and content. A statistical gain is obtained by taking advantage of the burstiness and variations in the video traffic.

While VBR coding in the ATM environment offers a number of advantages over fixed-rate coding in STM circuit-switched networks, it also causes problems that do not exist in STM networks. These problems are due to *cell loss* and *delay jitter*. Cell loss seems to be the major drawback for VBR video coding in ATM networks [13, 14] due to the high bit-rate and bursty nature of the traffic. Most research in ATM networks conducted today (call admission, bandwidth allocation, policing, scheduling) aims in fact at keeping the number of lost cells at a minimum, while admitting the maximum number of possibly diverse connections. However, owing to the asynchronous nature of the ATM concept, cell loss cannot be completely eliminated from the scene. Therefore, applications must be prepared to cope with this problem.

2 Compensative measures to tackle cell loss in ATM

The most natural compensative measure for cell loss is Automatic Repeat ReQuest (ARQ) [2], which simply requests retransmission of the missing (or corrupted) cells from the source. Although ARQ schemes are adequate for data services and other non-real-time applications, they are ill-suited for video transmission owing to large delays. Moreover, if the cell loss is due to congestion, the ARQ approach is likely to increase the congestion, as it effectively increases the volume of traffic offered to the network.

In terms of pure information content, a video session can usually accept some loss of information without impairing the image quality beyond a subjectively imperceptible or tolerable level. However, due to the propagative nature of VBR coding schemes, a single missing (or corrupted) cell may render a number of other (correctly received) cells unusable, because the information coded in them cannot be interpreted without the missing portion. This problem can be solved by periodically sending a frame containing the complete information about the image (i.e., a frame that can be interpreted without reference to the previous frames). This scheme, aimed at containing the size of the video segment affected by a lost or damaged cell, is called *enforced command refreshment.*

The third technique is *layering.* With this approach, image information is organized into several classes (layers) with different priorities. A lost cell may affect the quality of the image to a certain extent, depending on the priority of the layer to which the cell belongs. As low priority layers are intended to carry less relevant information than higher priority layers, it is postulated that in case of congestion, cells belonging to low-priority layers be dropped first.

ATM has a built-in mechanism for handling two cell priority levels. This suggests that compressed video information should be layered into high-priority (HP) and low-priority (LP) tiers [7, 8, 9, 11]. Our proposed scheme has been devised along these lines. It guarantees a minimum level of subjective performance at the decoder under conditions of extreme congestion, even when all low-priority cells are continuously being lost. It is also well suited for multicast sessions, where the same image must be sent over links with different capacity. By trimming the transmitted number of low priority cells to the available bandwidth, the router can make sure that all multicast recipients receive the same image with the quality of service commensurate with their expectations.

3 Layered source coding

Layered coding schemes can be broadly divided into those operating in the spatial domain (i.e., identifying different areas of relevance within the image) and those that function in the frequency domain (i.e., identify different levels of detail within the entire image). Our proposed coding method (called SPAFLAY, which stands for SPAtial and Frequency LAYering) combines these two approaches. In the spatial domain, a technique called Feature Plane Separation is used. In the frequency domain, we use a Combined Transform Coding scheme.

Feature Plane Separation (FPS). This is a content driven representation of video signals, normally done in the spatial domain, where we separate the image into areas of varying interest. For example, in a radiological environment, the tumor location on a chest x-ray might be more important to the physician than the surrounding area. In that case, we can send the cells from the region of interest with a higher priority than the cells from the background.

Combined Transform Coding (CTC). Video scenes typically contain a substantial amount of redundancy in the spatial/temporal domain. In other words, neighboring areas of a static image tend to be correlated, and the subsequent frames of a movie tend to differ only slightly. The purpose of transform coding is to map the highly correlated video data to a domain in which image coefficients become statistically independent.

Among the popular transforms useful for CTC are Cosine Transform (DCT) [12] and Haar Transform (HT) [3]. The transform coefficients are divided into two planes: the Upper Image Plane (UIP) and the Lower Image Plane (LIP) [5]. The UIP contains the most important information: error-free and timely delivery of UIP cells should be guaranteed. The LIP contains less important information, and the delivery of LIP cells is not always guaranteed. In case of congestion, some of the LIP cells may be lost. The advantage of this method is that even with the loss of all LIP cells, we are able to obtain a picture of acceptable quality using solely the UIP cells.

4 The rationale of SPAFLAY

Teleconferencing applications, including distance education, are usually characterized by a single area of interest (*fovea*) surrounded by a relatively unimportant background. This area of interest may be a "talking head," a location on the blackboard pointed to by the instructor, or an animated example/exhibit/experiment. It is important for the fovea to be received with a high fidelity, but even a significant deterioration in the quality of the background scene is usually tolerable. This suggests that a combination of FPS and CTC may be well suited for this kind of application.

The position of the fovea is dynamically changed to correspond to the current point of interest in the scene. A circular area surrounding the fovea is considered subjectively important. Information within this area is transform coded and sent in HP ATM cells. The background information is sent in two bursts: the low frequencies are sent as HP cells and the high frequencies as LP cells. The result of this scheme is a reduction of the subjective deterioration of the resultant image in the event of cell loss in the ATM network.

The LP cells describe the high frequency coefficients, which correspond to the "details" in the background portion of the image. In case of congestion, these cells will be lost first, which will result in a quality degradation of the background scene. But the background scene will appear continuous and undamaged, even if all LP cells are lost: it will only lose details and become "blocky."

Using the technique of transform coding alone does not give the advantage of specifying a spatial area of importance. Consequently, we are unable to guarantee acceptable quality for the critical portion of the image. On the other hand, if FPS is used all by itself, it may result in some fragments of the background being completely lost. The resultant image quality will be poor and irritating to the viewer. But when the two approaches are put together, we get a combination of their advantages and each of them absorbs the weak spots of the other. Notably, the combination of FPS and CTC also reduces the redundancy of the layered structure, which has remained one of the drawbacks in traditional layered coding algorithms. If the image happens to have no center of interest, we can still use CTC alone to good effect.

5 Stages in the SPAFLAY video codec

The coding algorithm is built around the following stages:

Block analysis

The objective of this stage is to reduce the temporal redundancy. The current frame is compared to the previous frame, looking for the image portions that have changed. The image is divided into blocks of 8x8 pixels and the corresponding blocks of the two frames are compared to detect areas of motion. Only the areas (blocks) which have "changed significantly" in the current frame with respect to the previous one will be compressed and transmitted.

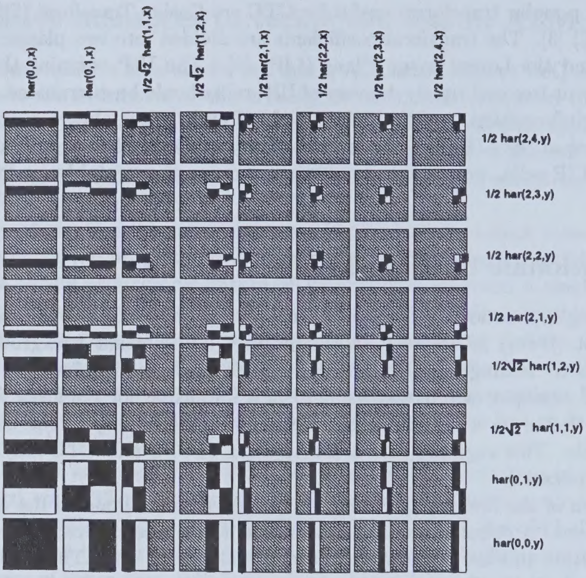


Figure 1: Two-dimensional Haar basis matrices for $N = 8$

The differencing step can be done quickly, reducing not only the bandwidth required to send video information but also the total amount of work needed to compress the frame, since only the areas that change need to be processed by the subsequent compression steps. The scheme to detect areas of motion is one of the unique features of SPAFLAY. In a teleconference scenario, it is reasonable to expect more motion in the area around the fovea. Limited motion in the background is not particularly important to the viewer. Hence SPAFLAY is more sensitive to motion around the fovea. This is achieved by having a variable threshold to detect motion blocks. The threshold values follow a parabolic curve, such that blocks which are closer to the fovea have a lower threshold.

Transform coding and layering

Each motion block marked in the previous stage is compressed using transform coding. The transform is a two-dimensional Haar Wavelet [1, 3]. The two-dimensional basis matrices for $N = 8$ are schematically presented in Fig. 1 [6].

Input to the Forward Haar Transform is an 8x8 motion block viewed as a 64-point discrete function of the two spatial dimensions x and y. The Forward Haar Transform takes these points as its input and converts them into 64 Haar Transform coefficients.

The purpose of layering is to assign priorities to different portions of the information representing the image. According to what we said earlier, this is done in two domains.

Spatial layering. The fovea position is dynamically altered to correspond to the point of interest in the scene. This is accomplished by assigning the fovea position to the block which was labeled as a motion block for the maximum number of times over an interval of 15 frames.

At the start of the video session, the initial position of the fovea is in the center of the image. The area of the circle describing the region of spatial importance around the fovea is input as a parameter.[1] Blocks of the video frame that fall within this region are considered to be high priority blocks, while the blocks that lie outside the spatial area of importance are deemed low priority blocks. If one of the high priority blocks has been marked as a motion block in the **Block analysis** step, it will be subjected to the full compression procedure. This involves performing a Forward Haar Wavelet Transform on the block. The resultant transform coefficients are then thresholded, sequenced in a zig-zag fashion, run-length encoded and transmitted in ATM cells with the CLP bit set to zero, which signifies high priority data.

---

[1] This area is specified as a percentage of the entire image.

Frequency layering. The blocks that have fallen outside the dynamic area of interest are low priority blocks. These blocks are encoded with two-level frequency splitting. The information representing the low-frequency component is sent as high-priority cells (with CLP set to zero). The high-frequency component is sent as low priority cells (CLP equal to one). We allow the user to specify the threshold separating "low" and "high" frequencies at the start of the video session as a **Quality** parameter quantized into 7 discrete levels (0–6).

With each of the 64 Haar Transform coefficients we associate a *Quality-Index* (QI) parameter, in the following way: First, the Haar basis vector 2 and the Haar basis vector 3 (Fig. 1) are treated as a single unit. Similarly, basis vectors 4, 5, 6, and 7 are treated as one unit. The other vectors are considered to be separate units. The idea is to group together vectors that are only locally significant. The 8x8 two-dimensional basis matrix (Fig. 1) is viewed as a 4x4 matrix of vector units. Each HT coefficient corresponding to an element in the 4x4 basis matrix is assigned a *Quality-Index* which is equal to the sum of the row and the column number of the element in the 4x4 matrix. This labeling is then carried over to the elements of the original 8x8 basis matrix. This way we have effectively labeled each of the 64 HT coefficients with a *Quality-Index.*

The *Quality-Index* of each HT coefficient is compared with the **Quality** value specified by the user. HT coefficients whose *Quality-Index* is less than or equal to **Quality** are separated from other HT coefficients of the same block. These coefficients are subjected to further compression steps and then transmitted as high priority data by packing them in ATM cells with the CLP bit set to zero. The remaining HT coefficients, whose *Quality-Index* is greater than **Quality**, are also subjected to further compression steps and then transmitted as low priority data, in ATM cells with the CLP bit set to one.

At the decoder end, we try to reconstruct the HT block by merging the low frequency components and the high frequency components. The resultant block is subjected to a Reverse Haar Transform. If the high frequency components were lost due to network congestion, the low frequency components alone go through this procedure.

Thresholding, sequencing, and run-length encoding

The input to this step is a block of 64 HT coefficients. The purpose of thresholding is to achieve further compression by representing HT coefficients with no greater precision than is necessary to achieve the desired image quality. The goal of this step is to discard information that is not visually significant by suppressing low-energy wavelet terms. In our psycho-visual experiments, we have found that HT coefficient values between -2 and 2 can be suppressed without any significant loss in the decoded image quality.

The thresholded coefficients are ordered in a zig-zag sequence, as shown in Fig. 2. This re-ordering facilitates entropy coding techniques, like run-length encoding, by placing low-frequency coefficients (which are more likely to be nonzero) before high-frequency coefficients. It is to be noted that the zig-zag sequence is arranged such that all the HT coefficients of a given *Quality-Index* value are packed and sent together. Also, these HT coefficients are sent in the increasing order of *Quality-Index.* With this ordering, entire groups of HT coefficients corresponding to a particular *Quality-Index* value are reduced to zeros.

Finally, the HT coefficients are subjected to run-length encoding before being transmitted. In this step, each nonzero coefficient is represented in combination with the number (run-length)

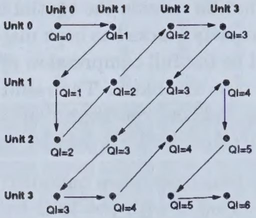


Figure 2: Zig-zag sequencing of Haar Transform coefficients

of zero-valued coefficients that follow it in the zig-zag sequence. The run-length encoded samples are transmitted over the network.

Command refreshment

To improve the image quality and to prevent errors in VBR coding from propagating, SPAFLAY uses a form of command refreshment. Portions of the frame are transmitted every once in a while irrespective of their motion content. These are blocks detected as stationary over many consecutive video frames. They are transform coded but not thresholded, i.e., they are sent in high resolution. This process results in a nearly lossless version of the original image, while still achieving approximately 2:1 compression.

One important role of command refreshment in SPAFLAY is in multicast applications. If a client joins a multicast session in progress, it has no previous image data into which the differential motion blocks arriving from the server can be embedded. Consequently, without a form of command refreshment, the client could have to wait for a long time to fill the entire image.

6 Bit-rate-penalty in SPAFLAY

The layered coding algorithm used in SPAFLAY increases the robustness of the video codec in the face of cell loss. However, this advantage comes at the expense of incurring a bit-rate-penalty compared to single-layer approaches. This is due to an increase in the redundancy of the layered structure, which comes from two sources:

- Video blocks subjected to frequency splitting are sent in two separate bursts: low frequencies of the block sent as high priority data and high frequencies of the same block sent as low priority data.
- At the encoding end, the **decoded** video frame needs to be used as a reference to detect the motion blocks for the next captured frame. If the quality of the decoded image is poor, then besides the actual motion blocks, blocks that are "coarse" will also be spuriously detected as motion blocks. Hence, more "motion blocks" are transmitted than actually necessary.

To measure the extent of bit-rate-penalty in SPAFLAY, we can compare the amount of information corresponding to the same sequence of scenes with different parameters of the algorithm. In particular, SPAFLAY has a *cut-through* mode of operation in which neither spatial nor frequency layering is used. A video sequence was filmed and compressed using various options in SPAFLAY. In each case, the proportion of high priority and low priority data in the compressed stream is reflected in the quality of the decoded image when all the low priority data gets lost. This is shown in Figures 3–12. The "Spatial" percentage gives the relative size of the area of importance surrounding the fovea.

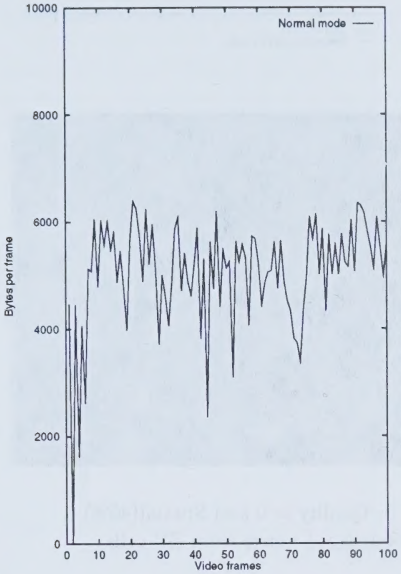


Figure 3: Case: cut-through mode

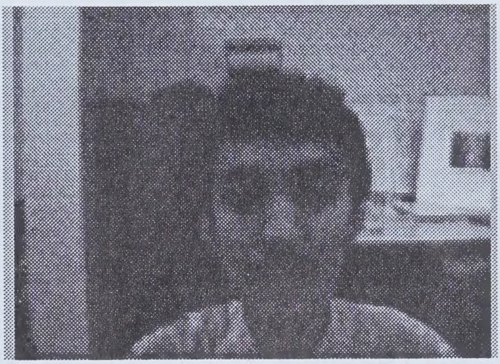


Figure 4: Cut-through mode: decompressed image

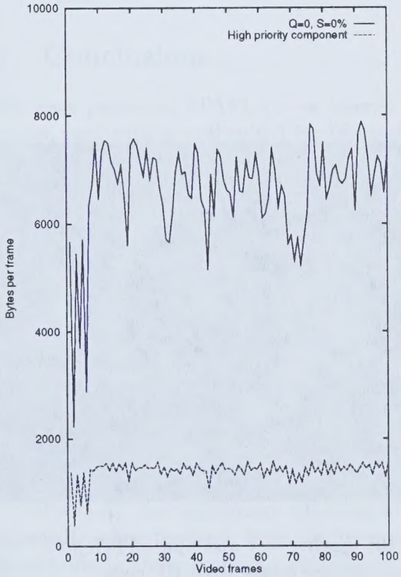


Figure 5: Case: Quality = 0 and Spatial (0%)

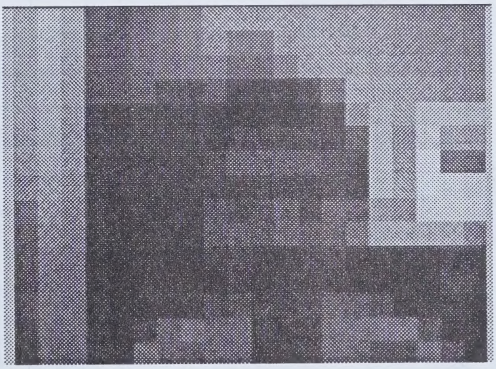


Figure 6: Quality = 0 and Spatial (0%): decompressed image from HP cells

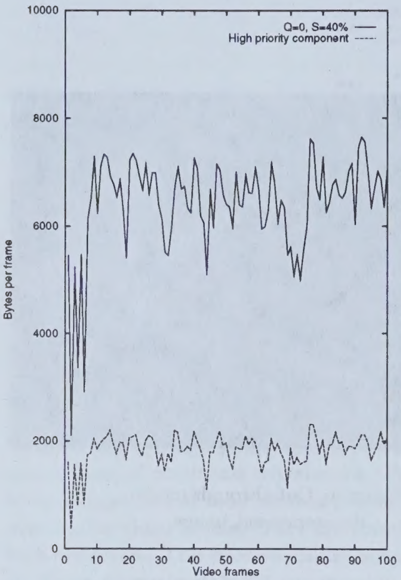


Figure 7: Case: Quality = 0 and Spatial (40%)

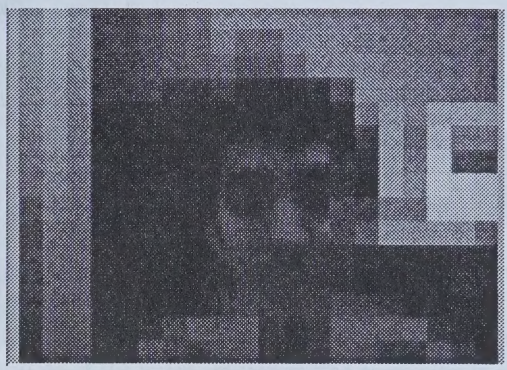


Figure 8: Quality = 0 and Spatial(40%): decompressed image from HP cells

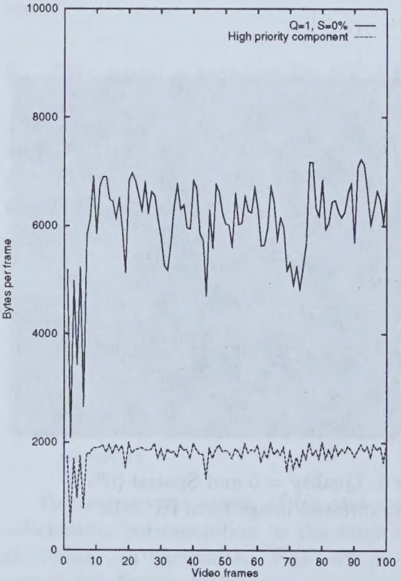


Figure 9: Case: Quality = 1 and Spatial (0%)

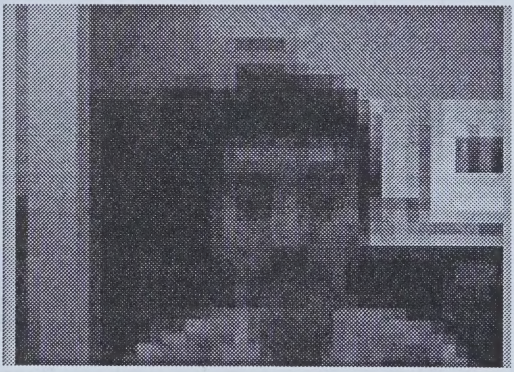


Figure 10: Quality = 1 and Spatial (0%): decompressed image from HP cells

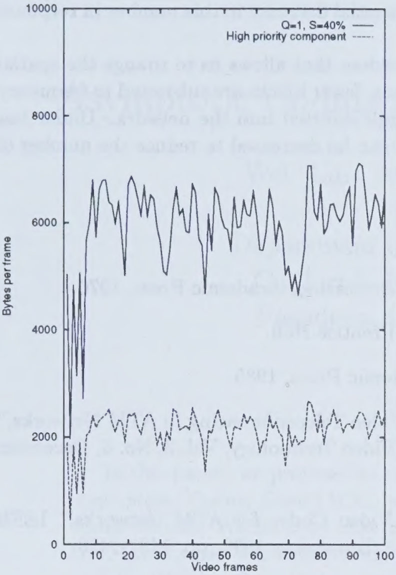


Figure 11: Case: Quality = 1 and Spatial (40%)

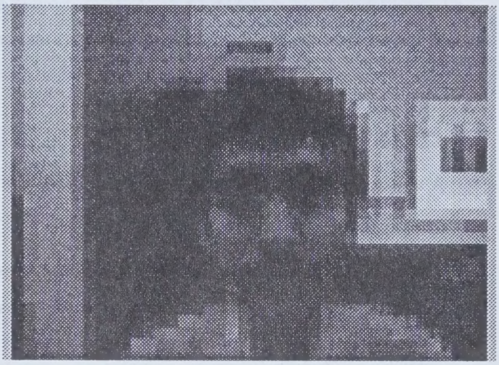


Figure 12: Quality = 1 and Spatial (40%): decompressed image from HP cells

7 Conclusions

We have presented SPAFLAY—a layered coding algorithm for motion video over ATM networks, particularly well suited for teleconferencing and distance education. The talking-head scene in these applications provides an ideal fovea location, which favors the spatial layering in SPAFLAY.

The automated prioritization scheme in SPAFLAY can be used to good advantage in a multicast session, where some clients are connected to the image server through low-speed links, while others are connected via high-speed links. Instead of forcing everyone to use the same lowest-common-bandwidth, reduced quality video encoding, the SPAFLAY codec will distribute to each client the number of cells that the client's link can comfortably accommodate. This will automatically deliver the best possible image quality that can be achieved with the available bandwidth.

Although the two-priority levels built into SPAFLAY are tailored to ATM, our algorithm can be used with any other network offering two or more packet priority levels. In fact, SPAFLAY can be easily modified to accommodate more than two priorities. We are currently working on these modifications and also on extending SPAFLAY to scenes with multiple fovea.

SPAFLAY has significant advantages over techniques that use frequency layering alone, especially when feedback from the network is available and can be used to alter the HP/LP breakpoint in the compressed video stream. Layering techniques based on frequency splitting alone result in a higher number of transform coefficients sent as HP data when the network is not congested. With these schemes, it is postulated that the session should start with a high proportion of HP data, possibly penalizing other less time-critical traffic. When feedback from the network indicates that HP data is likely to be lost due to congestion, the source should dynamically decrease the number of transform coefficients sent as high priority data. There exist arguments in favor of a linear increase and exponential decrease in this number in response to the congestion feedback.

In SPAFLAY, we have an additional degree of freedom that allows us to change the spatial area of importance. As the area of importance increases, fewer blocks are subjected to frequency splitting, which results in a higher number of HP data inserted into the network. Under less favorable network conditions, the area of importance can be decreased to reduce the number of HP cells.

References

[1] Andrews H.C., *Computer Techniques in Image Processing,* Academic Press, 1970.

[2] Bertsekas D., and Gallager R., *Data Networks,* Prentice-Hall.

[3] Clarke R.J., *Transform Coding of Images,* Academic Press, 1985.

[4] Cohen D.M., et.al., "Performance Modeling of Video Teleconferencing in ATM Networks," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 3, No. 6, December 1993, pp. 408-420.

[5] Ghanbari M., "An adapted H.261 Two-Layer Video Codec for ATM Networks," IEEE Transactions on Communications, Vol. 40, No. 9, September 1992, pp. 1481-1490.

[6] Hall E.L., *Computer Image Processing and Recognition,* Academic Press, 1979.

[7] Hosoda K., "Variable Rate Video Coding Scheme for Broadcast Quality Transmission and its ATM Network Applications," IEICE Trans. Commun., Vol. E75-B, No. 5, May 1992, pp. 349-357.

[8] Joseph K., et.al, "Design and Performance Evaluation of a Variable Bit-Rate (VBR) Video Compression Algorithm for ATM Networks," GLOBECOM '91, pp. 9-15.

[9] Kinoshita T., et.al., "Variable-Bit-Rate HDTV CODEC with ATM-Cell-Loss Compensation," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 3, No. 3, June 1993, pp. 230-237.

[10] Kishimoto R., et.al., "Generation Interval Distribution Characteristics of Packetized Variable Rate Video Coding Data Streams in an ATM Network," IEEE Journal on Selected Areas in Communications, Vol. 7, No. 5, June 1989, pp. 833-841.

[11] Kishino F., et.al., "Variable Bit-Rate Coding of Video Signals for ATM Networks," IEEE Journal on Selected Areas in Communications, Vol. 7, No. 5, June 1989, pp. 8801-806.

[12] Pennebaker W.B., and Mitchell J.L., *JPEG Still Image Data Compression Standard.* Van Nostrand Reinhold.

[13] Verbiest Willem., "A Variable Bit Rate Video Codec for Asynchronous Transfer Mode Networks," IEEE Journal on Selected Areas in Communications, Vol. 7, No. 5, June 1989, pp. 761-770.

[14] Zhang Y.Q., et.al, "Variable-Bit-Rate Video Transmission in Broadband-ISDN Environment," IEEE Trans. Commun., pp. 213-222.

Asymptotic Coding Gain and Trellis Complexity*

Wei Lin and Robert J. McEliece

Department of Electrical Engineering
California Institute of Technology
Pasadena, California 91125, USA

Abstract

In this paper, we propose the ratio of *Logarithmic Trellis Complexity* (LTC) to *Asymptotic Coding Gain* (ACG) as a measure of the efficiency of codes, block or convolutional. We show by examples and theorems that, for any linear code, the LTC–ACG ratio is always strictly larger than one, a result which is similar to one in [14]. We also show that asymptotically the LTC–ACG ratio for any sequence of optimal convolutional codes with a fixed n and k must exceed 2, and present an algorithm useful for computing the LTC for convolutional codes.

1 Introduction

It is well–known that linear block codes have a natural minimal trellis structure which allows them to be decoded with the Viterbi algorithm [2, 7, 10, 11, 12, 13, 14, 15, 16, 17, 21]. Recently, V. Sidorenko and V. Zyablov [23] developed a theory of minimal trellises for convolutional codes based on the codes' parity-check matrices. Shortly afterwards, the authors of this paper independently discovered a similar theory, this time based on the codes' generator matrices [20]. In any case, the computational complexity of the Viterbi decoding algorithm for either block or convolutional codes can be accurately estimated by the number of trellis edges per encoded bit [16, 17, 20]. Since this number is typically large, we define *logarithmic trellis complexity* (LTC) of a code as the base–2 logarithm of the number of trellis edges per encoded bit. In this paper, we take the LTC as the measure of coding complexity.

On the other hand, the *Asymptotic Coding Gain* (ACG) is a widely accepted measure of the performance of a coding system. The ACG is defined as the product of code's rate and minimal (block codes) or free (convolutional codes) distance. The ACG measures the coding gain of the code when it is used on an AWGN channel and the signal-to-noise ratio is very large. Thus the larger the ACG, the better the code. Unfortunately, codes

---

*This work was partially supported by NSF grant no. NCR-9505975 and a grant from Pacific Bell.

with large ACG's usually also have high LTC's, and so it is natural to consider the LTC–ACG ratio.

In this paper, we shall examine the relationship between LTC and ACG for both block codes and convolutional codes. Section 2 addresses block codes. Since it is difficult to compute the LTC-ACG ratio for most codes, we present a universal lower bound, which is similar to a result in [14]. Emprical comparisons among block codes are included in Section 2. In Section 3, we consider convolutional codes. We begin by presenting an algorithm for producing the "minimal trellis matrix" for any convolutional code. Then we use block codes to "approximate" convolutional codes, which allows us to prove that the LTC–ACG ratio for any sequence of optimal convolutional codes with fixed n and k must exceed 2.

2 Block Codes

2.1 Preliminaries

Let V_n be the n–dimensional vector space over the finite field $GF(2)$. Given a vector $x = (x_0, x_1, x_2, \ldots, x_{n-1}) \in V_n$, let $L_x = \min\{i : \ x_i \neq 0\}$ and $R_x = \max\{i : \ x_i \neq 0\}$. The number $m_x = R_x - L_x + 1$ is called the *span* of the vector x.

A generator matrix $G^T = (r_0, r_1, r_2, \ldots r_{k-1})$ over V_n of a linear $[n, k, d]$ code is called a *minimal span generator matrix* (MSGM) if and only if it satisfies the "LR" property, defined as follows:

$$(a) \quad L_{r_i} \neq L_{r_j} \ \text{ if } \ i \neq j; \qquad (b) \quad R_{r_i} \neq R_{r_j} \ \text{ if } \ i \neq j$$

The number $m_G = \sum_i m_{r_i}$ is called the *span* of G [16, 17].

Each code has a unique minimal trellis structure, which can be constructed from a MSGM. In the minimal trellis, each vertex is assigned a *depth* in $\{0, 1, \ldots, n\}$, which is its distance from the root. For any $i, j \in \{0, 1, \ldots, n\}$, $j \neq n$, define the *state complexity* s_i at depth i and the *edge complexity* e_j at depth j to be

$$s_i = \log_2(\text{number of vertices at depth } i)$$

$$e_j = \log_2(\text{number of edges connecting vertices at depth } j \text{ and } j+1).$$

It follows that

$$s_i = k - f_i - p_i; \qquad e_i = k - f_{i+1} - p_i \tag{1}$$

where $f_i = |\{j : \ L_{r_j} \geq i\}|$, and $p_i = |\{j : \ R_{r_j} < i\}|$.

The most important application of trellises for block codes is to facilitate the implementation of the Viterbi algorithm. In [16], it was shown that a good measure of the computational complexity of the Viterbi algorithm for block codes is $|E|$, the number of edges in the trellis. Since after decoding, we have estimates for k information bits, we define the *logarithmic trellis complexity* of a block code by

$$LTC = \log_2 \frac{|E|}{k} \tag{2}$$

On the other hand, the performance of a code is often measured by the *asymptotic coding gain*, which is defined as

$$ACG = \frac{k}{n}d \tag{3}$$

The rest of the paper is devoted to a study of the relationship of LTC to ACG.

2.2 Lower bound on LTC-ACG ratio

To derive a lower bound on the LTC–ACG ratio, it is necessary to lower bound the total number of edges in the minimal trellis in terms of n, k, and d. This can be done by studying the properties of a MSGM.

Lemma 1: For any $[n, k, d]$ linear code, $m_G \geq kd$.

Proof: Each row of the MSGM is itself a non–zero codeword. There are at least d non–zero entries in each such codeword. Hence $m_{r_i} \geq d$. Summing over all rows leads to the conclusion. ♢

Lemma 2: For any $[n, k, d]$ linear code, $\sum e_i = m_G$

Proof: An entry a_{ij} in the generator matrix is called *active* if $L_{r_i} \leq j \leq R_{r_i}$. We shall count the number of active entries in two ways. The number of active elements in row r_i is $|\{j: \ L_{r_i} \leq j \leq R_{r_i}\}|$, which is m_{r_i}. Thus the total number of active elements is the sum of the spans, i.e. m_G.

On the other hand, the number of active elements in column j is

$$\begin{aligned} |\{i : L_{r_i} \leq j \leq R_{r_i}\}| &= k - |\{i : j+1 \leq L_{r_i}\}| - |\{i : j > R_{r_i}\}| \\ &= k - f_{j+1} - p_j = e_j \end{aligned}$$

Thus $\sum_i e_i = m_G$. ♢

Theorem 3: For any linear code

$$LTC \geq ACG - \log_2 R. \tag{4}$$

Proof: By the arithmetic–geometric inequality,

$$\begin{aligned} |E| &= \sum_i 2^{e_i} \geq n(2^{\sum_i e_i})^{\frac{1}{n}} \\ &\geq n2^{\frac{m_G}{n}} \quad \text{(By Lemma 2)}. \end{aligned} \tag{5}$$

Hence

$$\begin{aligned} LTC = \log_2 \frac{|E|}{k} &\geq \log_2 \frac{n}{k} 2^{\frac{m_G}{n}} \geq \log_2 \frac{n}{k} 2^{\frac{kd}{n}} \quad \text{(By Lemma 1)} \\ &= \log_2 \frac{n}{k} + \frac{kd}{n} = ACG - \log_2 R. \end{aligned} \tag{6}$$

♢

Corollary: $LTC \geq ACG$, with equality iff $R = 1$.

Example 1: The $[n, n-1, 2]$ parity–check code.

Its minimal trellis has $4(n-1)$ edges. Thus $LTC = 2$. The ACG is $2\frac{n-1}{n}$. Then $\frac{LTC}{ACG} = \frac{n}{n-1}$. Note that this ratio is always strictly greater than 1 but it approaches 1 as the code length goes to infinity. Hence (4) is asymptotically tight in this case.

Example 2: The $[n, 1, n]$ repetition code.

The ACG is 1 but $LTC = 1 + \log_2 n$. Here the LTC–ACG ratio is $1 + \log_2 n$. Since $R = 1/n$, the bound (4) is exact in this case.

Example 3: The first order RM codes.

In [10, 11], the optimal trellises for Reed-Muller codes were completely determined. For the first order RM codes, $n = 2^m, k = m+1, d = 2^{m-1}$. Then $ACG = \frac{m+1}{2}$. A computation of total span gives $m_G = (m-1)2^m + 2$. From (5), $LTC \geq \frac{(m-1)2^m+2}{n} - \log_2 R = m - 1 + \frac{2}{n} + m - \log_2(m+1)$. Thus

$$\lim_{n\to\infty} \inf \frac{LTC}{ACG} \geq 4$$

but (4) gives 3 as the lower bound.

From (1), we have $\sum_{i=0}^{n} s_i = nk + k - \sum_{i=0}^{n} f_i - \sum_{i=0}^{n} p_i = nk + k - \sum_{i=1}^{n} f_i - \sum_{i=0}^{n-1} p_i - k - k = m_G - k$. Let $s_{max} = \max\{s_i\}$. Then $(n-1)s_{max} \geq \sum_{i=0}^{n} s_i = m_G - k \geq kd - k$, i.e., $s_{max} \geq \frac{k(d-1)}{n-1} > \frac{k(d-1)}{n}$. This is Theorem 3 in [15], with a simple proof.

To study the asymptotic behavior of the LTC–ACG ratio, we first note that in [14], some asymptotic results on s_{max} were presented. These results are closely related to the discussion in this paper because they imply that for any sequence of block codes with a fixed rate $R > 0$ and fixed relative distance $\delta = \frac{d}{n} > 0$, as n goes to infinity,

$$\lim_{n\to\infty} \inf \frac{LTC}{ACG} \geq 2 \tag{7}$$

However, This result does not cover *Example 1* to *Example 3* and many popular codes, including higher order RM and BCH codes. Combining (4) and (7) gives the following asymptotic lower bound for linear codes.

Theorem 4: For any sequence of linear codes,

$$\lim_{n\to\infty} \inf \frac{LTC}{ACG} \geq \begin{cases} 2 & \text{if } R\delta > 0 \\ 1 & \text{if } R\delta = 0 \end{cases} \tag{8}$$

3 Convolutional codes

3.1 The minimal trellise for a convolutional code

An (n, k) binary convolutional code is a k-dimensional subspace of $F(D)^n$, where $F(D)$ is the field of rational functions in D over $GF(2)$. Any (n, k) convolutional code is generated by a minimal matrix polynomial (or equivalently by a minimal polynomial matrix) $G(D) = G_0 + G_1 D + G_2 D^2 + \ldots + G_L D^L = (G_{i,j}(D))$, where G_i is a $k \times n$ binary scalar matrix, and $G_{i,j}(D)$ is a polyomial . Without loss of generality, we can assume G_0 and G_L are both non–zero. In modern coding parlance, $\max_j\{$ degree of $G_{i,j}(D)\}$ is called the *Forney index* of row i. Let the Forney indices be denoted $e_1 \leq e_2 \leq \ldots \leq e_k$. Then the largest Forney index $e_k = L$ is the memory order.

Trellises have been used for convolutional codes for about three decades. Surprisingly, however, the relatively recent developement of trellis theory for block codes has lead to some new insights about trellises for convolutional codes. To see why this is so, it is convenient to introduce the discrete semi-infinite generator matrix G^* for convolutional codes.

$$G^* = \begin{bmatrix} G_0 & G_1 & G_2 & \dots & G_L & & & & \\ & G_0 & G_1 & G_2 & \dots & G_L & & & \\ & & G_0 & G_1 & G_2 & \dots & G_L & & \\ & & & G_0 & G_1 & G_2 & \dots & G_L & \\ & & & & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}$$

According to the trellis theory for block codes, if there are e active elements in one column of G^*, the number of edges in the corresponding stage of the trellis is 2^e. If the number of active elements can be reduced, the total number of trellis edges can also be reduced.

Based on these simple facts, a theory of minimal trellises for convolutional codes was developed in [20]. It was shown that for every convolutional code, there is a unique minimal trellis structure consisting of repeated copies of a *trellis module*, except for a short initial transient block. Each trellis module corresponds to one vertical section of G^*, i.e.,

$$\hat{G} = \begin{pmatrix} G_L \\ G_{L-1} \\ \vdots \\ G_1 \\ G_0 \end{pmatrix}$$

This trellis is minimal in the sense that it has the fewest number of edges per information bit among all trellis realizations for the same code. G^* is called a *minimal trellis matrix* if and only if for any two different rows of G^*, r_i and r_j,

$$\text{(a)}\ L_{r_i} - L_{r_j} \neq 0 \mod n; \qquad \text{(b)}\ R_{r_i} - R_{r_j} \neq 0 \mod n. \tag{9}$$

The conditions in (9) are called the "LR property" for convolutional codes.

In analogy with the situation for block codes, the LTC for a convolutional code is defined as in (2), where now $|E|$ is the number of edges in a minimal trellis module. In [20], it is shown that further reduction in the LTC may be achieved if we allow column permutations of the polynomial generator matrix.

3.2 An algorithm for computing a MSGM for a convolutional code

The question now arises as to how to produce the minimal trellis matrix for a given convolutional code. Fortunately, there is a simple algorithm which always produces the minimal trellis matrix. In what follows, assume $i < j$ and let the j^{th} row of the polynomial generator matrix $(G_{i,j}(D))$ be denoted by Row_j.

·∞· An algorithm for finding a minimal trellis matrix for a convolutional code ·∞·

```
Begin:
If (LR property does not hold)
{
    If (($L_{r_i} = L_{r_j} \mod n$ and $R_{r_i} \geq R_{r_j}$ ) or ($R_{r_i} = R_{r_j} \mod n$ and $L_{r_i} < L_{r_j}$ ))
    {
        $t = (j-i) \mod k$; $s = \frac{j-i-t}{k}$; $i = i \bmod k$; $j = j \bmod k$;
        $Row_i = Row_j D^s + Row_i$
    }
    If (($L_{r_i} = L_{r_j} \mod n$ and $R_{r_i} < R_{r_j}$ ) or ($R_{r_i} = R_{r_j} \mod n$ and $L_{r_i} \geq L_{r_j}$ ))
    {
        $t = (j-i) \mod k$; $s = \frac{j-i-t}{k}$; $i = i \bmod k$; $j = j\ \mathit{bmod}\ k$;
        $Row_j = Row_j + Row_i D^{-s}$
    }
}
End.
```

Theorem 5: The algorithm is convergent.
Proof: Without loss of generality, assume $L_{r_i} = L_{r_j} \mod n$, where $i < j$. Let $t = (j-i) \mod k$, $s = \frac{j-i-t}{k}$, $m = i \bmod k$, $n = j \bmod k$. If $R_{r_i} \geq R_{r_j}$, adding r_j to r_i will reduce the total span in G^*. If $R_{r_i} < R_{r_j}$, adding r_i to r_j will also reduce the total span in G^*. Since this procedure reduces the total number of active elements in G^*, it will terminate after a finite number of steps.

3.3 The LTC-ACG ratio

The sth truncation of an (n, k, m) convolutional code is an $[n(e_k + s), ks]$ block code whose generator matrix G^s is just the first ks rows of G^*. Denote the sth truncation of the code C by C^s. If G^* is a minimal trellis matrix for the convolutional code, then it is easy to see that G^s is a MSGM of C^s.

Lemma 6:

$$\lim_{s \to \infty} LTC_{C^s} = LTC_C$$

Proof: If s is large enough, the minimal trellis for C^s contains $s - e_k$ copies of the trellis module, in addition to a head and a tail whose matrix forms are:

$$G_{head} = \begin{bmatrix} G_0 & G_1 & G_2 & \dots & G_{L-1} \\ & G_0 & G_1 & \dots & G_{L_2} \\ & & \dots & \dots & \dots \\ & & & & G_0 \end{bmatrix}; \quad G_{tail} = \begin{bmatrix} G_L & & & & \\ G_{L-1} & G_L & & & \\ \dots & \dots & \dots & \dots & \\ G_1 & \dots & G_{L-2} & G_{L-1} & G_L \end{bmatrix}$$

Then $LTC_{C^s} = \log_2 \frac{(s-e_k)|E| + E_{tail} + E_{head}}{ks}$ where E_{tail} and E_{head} are the number of edges in the tail and head, respectively. Since $0 < E_{tail} < e_k|E|$, $0 < E_{head} < e_k|E|$, $\log_2 \frac{(s-e_k)|E|}{ks} < LTC_{C^s} < \log_2 \frac{(s+e_k)|E|}{ks}$. Note that n, k and e_k are fixed. Let s go to infinity:

$$\lim_{s \to \infty} \log_2 \frac{s-e_k}{s} + \log_2 \frac{|E|}{k} < \lim_{s \to \infty} LTC_{C^s} < \log_2 \frac{|E|}{k} + \lim_{s \to \infty} \log_2 \frac{s+e_k}{s} \qquad (10)$$

$$\log_2 \frac{|E|}{k} \le \lim_{s\to\infty} LTC_{C^s} \le \log_2 \frac{|E|}{k} \diamond$$

Theorem 7: For any convolutional code C,

$$LTC \ge ACG - \log_2 R \tag{11}$$

Proof: Since every codeword of C^s is a codeword of C^{s+1} and C, $d^s \ge d^{s+1} \ge d_{free}$. From Theorem 3, $LTC_{C^s} \ge ACG_{C^s} - \log_2 \frac{ks}{n(e_k+s)}$. Thus

$$\begin{aligned} LTC = \lim_{s\to\infty} LTC_{C^s} &\ge \lim_{s\to\infty} \frac{k}{n} d^s - \log_2 R \\ &\ge ACG - \log_2 R \diamond \end{aligned}$$

In what follows, we shall investigate the asymptotic behaviour of the LTC–ACG ratio for optimal convolutional codes for fixed values of n and k, as the memory m approaches infinity. Figure 1 shows that for the best $(2, 1, m)$ convolutional codes, the LTC–ACG ratio approaches, and then crosses, a line of slope 2, as the memory order increases. In fact, we can prove that this is true for any sequence of convolutional codes. To do this, we will continue to employ block codes to "approximate" convolutional codes.

Consider the sequence of block codes $C^s_{e_k}$, which are the sth truncations of an optimal (n, k, m) convolutional code C_{e_k}. If e_k and s approach infinity independently, we can hardly expect a meaningful result, so hereafter, let $s = \lambda e_k$ for some constant $\lambda > 1$. For this subsequence, the rate and relative distance are both convergent.

$$\lim_{e_k\to\infty} \frac{ks}{n(e_k+s)} = \frac{\lambda}{1+\lambda} \cdot \frac{k}{n} = \frac{\lambda}{1+\lambda} R > 0$$

$$\delta = \lim_{e_k\to\infty} \frac{d_{C^s_{e_k}}}{n(e_k+s)} = \frac{1}{n(1+\lambda)} \lim_{e_k\to\infty} \frac{d_{C^s_{e_k}}}{e_k} \ge \frac{1}{n(1+\lambda)} \lim_{e_k\to\infty} \frac{d_{free}}{e_k} > 0 \tag{12}$$

From Theorem 4, we have

$$\begin{aligned} \liminf_{e_k\to\infty} \frac{LTC_{C^s_{e_k}}}{n(e_k+s)} &= \frac{1}{n(1+\lambda)} \liminf_{e_k\to\infty} \frac{LTC_{C^s_{e_k}}}{e_k} \\ &\ge 2 \lim_{e_k\to\infty} \frac{ACG_{C^s_{e_k}}}{n(e_k+s)} \end{aligned} \tag{13}$$

Lemma 8:

$$\lim_{e_k\to\infty} \frac{LTC_{C^s_{e_k}} - LTC_{e_k}}{e_k} = 0 \tag{14}$$

Proof: From (10) we have:

$$\lim_{e_k\to\infty} \frac{\log_2 \frac{\lambda-1}{\lambda}}{e_k} < \lim_{e_k\to\infty} \frac{LTC_{C^s_{e_k}} - LTC_{e_k}}{e_k} < \lim_{e_k\to\infty} \frac{\log_2 \frac{\lambda+1}{\lambda}}{e_k}$$

i.e.,

$$|\lim_{e_k\to\infty} \frac{LTC_{C^s_{e_k}} - LTC_{e_k}}{e_k}| < \lim_{e_k\to\infty} \frac{c}{e_k} \to 0 \diamond$$

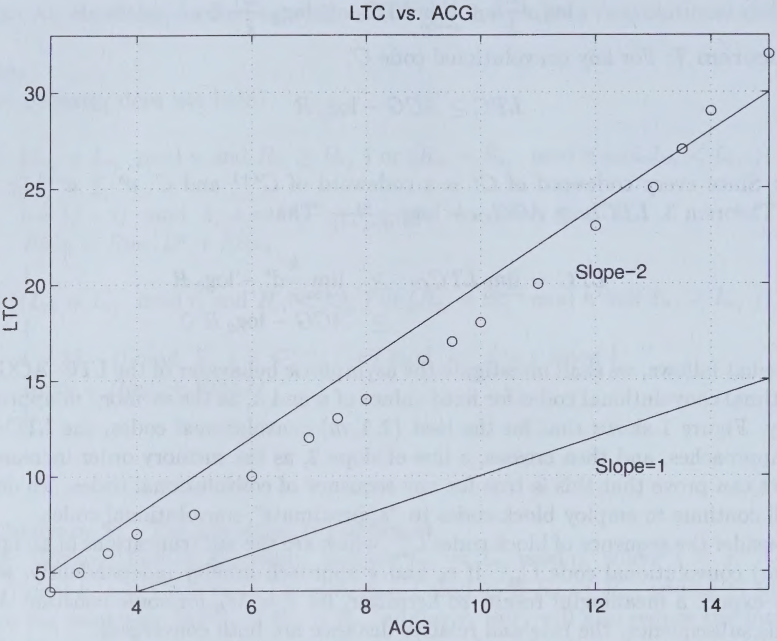


Figure 1. Best (2,1,m) convolutional codes with increasing memory order

Theorem 9: For any sequence of convolutional codes of increasing memory order e_k with fixed n and k,

$$\lim_{e_k\to\infty} \inf \frac{LTC_{e_k}}{ACG_{e_k}} \geq 2$$

Proof: From Lemma 8 and (12), (13)

$$\begin{aligned}
\lim_{e_k\to\infty} \inf \frac{LTC_{e_k}}{ACG_{e_k}} &= \frac{\lim_{e_k\to\infty} \inf \frac{LTC_{e_k}}{e_k}}{R \lim_{e_k\to\infty} \frac{d_{free}}{e_k}} = \frac{\lim_{e_k\to\infty} \inf \frac{LTC_{C^s_{e_k}}}{e_k}}{R \lim_{e_k\to\infty} \frac{d_{free}}{e_k}} \\
&\geq \frac{2n(1+\lambda) \lim_{e_k\to\infty} \frac{ACG_{C^s_{e_k}}}{n(e_k+s)}}{R \lim_{e_k\to\infty} \frac{d_{free}}{e_k}} \\
&= \frac{2\lambda}{1+\lambda} \frac{Rn(1+\lambda)\delta}{R \lim_{e_k\to\infty} \frac{d_{free}}{e_k}} \\
&\geq \frac{2\lambda}{1+\lambda} \frac{\lim_{e_k\to\infty} \frac{d_{free}}{e_k}}{\lim_{e_k\to\infty} \frac{d_{free}}{e_k}} \\
&= \frac{2\lambda}{1+\lambda}
\end{aligned}$$

The theorem follows from that λ can be arbitrarily large.◇

References

[1] K. Abdel–Ghaffar, R. J. McEliece, and G. Solomon, " Some Partial–Unit–Memory Convolutional Codes," *JPL TDA Progress Report,* vol. 42–107 (November 1991), pp. 57-72. A summary of this paper also appeared in *Proc. 1991 International Symposium on Information theory,* p. 196.

[2] L. R. Bahl, J.Cocke, F. Jelinek and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error rate," *IEEE Trans. Inform. Theory,* vol. IT–20, pp. 284–287, March 1974.

[3] G. C. Clark, Jr., and J. B. Cain, *Error-Correction Coding for Digital Communications.* New Yark: Plenum Press, 1981.

[4] D. J. Costello, Jr., "Free Distance Bounds for Convolutional Codes," *IEEE Trans. inform. Theory,* vol. IT–20, pp. 356–365, May 1974.

[5] D. G. Daut, J. W. Modestino, and L. D. Wismer, "New Short Constraint Length Convolutional Code Constructions for Selected Rational Rates," *IEEE Trans. Inform. Theory,* vol. IT–28, pp. 794–800, September 1982.

[6] A. Dholakia, *Introduction to Convolutional Codes with Applications.* Boston: Kluwer Academic Publishers, 1994.

[7] S. Dolinar, L. Ekroot, A. B. Kiely, W. Lin, and R. J. McEliece, " The Permutation Trellis Complexity of Linear Block Codes," Proc. 32nd Annual Allerton Conf. on Communication, Control, and Computing, (September 1994), pp. 60–74.

[8] G. D. Forney Jr., "The Viterbi Algorithm," Proc. IEEE, Vol 61, pp268–276, March, 1973.

[9] G. D. Forney Jr., "Convolutional Codes I: Algebraic Structure," *IEEE Trans. Inform. Theory,* vol. IT–16, pp. 268–278, November 1970.

[10] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "On the Optimum Bit Orders with Respect to the State Complexity of trellis Diagrams for Binary Linear Codes," *IEEE Trans. Inform. Theory,* vol. IT–39, pp. 242–245, January 1993.

[11] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "On Complexity of Trellis Structure of Linear Block codes," *IEEE Trans. Inform. Theory,* vol. IT–39, pp. 1057–1064, May 1993.

[12] A. B. Kiely, S. Dolinar, R. J. McEliece, L. Ekroot, and W. Lin, "Trellis Decoding Complexity of Linear Block Codes," submitted to *IEEE Trans. Inform. theory,* November 1994.

[13] F. R. Kschischang and V. Sorokine, "On the Trellis Structure of Block Code," submitted to *IEEE Trans. Inform. Theory.*

[14] A. Lafourcade, and A. Vardy, "Lower Bounds on Trellis Complexity of Block Codes," preprint, December, 1994.

[15] A. Lafourcade, and A. Vardy, "Asymptotically Good Codes Have Infinite Trellis Complexity," *IEEE Trans. Inform. Theory* vol. IT–41, pp. 555–559, March 1995.

[16] R. J. McEliece, " The Viterbi Decoding Complexity of Linear Block Codes," Proc. 1994 ISIT, p. 341.

[17] R. J. McEliece, " On the BCJR Trellis for Linear Block Codes," submitted to *IEEE Trans. Inform. theory.*

[18] R. J. McEliece, and R. P. Stanley, "The General Theory of Convolutional Codes," *JPL TDA Progress Report,* vol. 42–113 (May 1993), pp. 89–98.

[19] R. J. McEliece, E. R. Rodemich, H. Rumsey, JR., and L. R. Welch, " New Upper Bounds on the Rate of a Code via the Delsarte–MacWilliams Inequalities," *IEEE Trans. inform. Theory,* vol. IT–23, pp. 157–166, March 1977.

[20] R. J. McEliece and W. Lin, "The Trellis Complexity of Convolutional Codes," *Presented at 3rd International Symposium on Communication Theory and Application, U.K.,* July, 1995.

[21] D. J. Muder, " Minimal Trellises for Block Codes," *IEEE Trans. inform. Theory,* vol. IT–34, pp. 1049–1053, September 1988.

[22] E. Passke, " Short Binary Convolutional Codes with Maximal Free Distance for Rate 2/3 and 3/4," *IEEE Trans. inform. Theory,* vol. IT–20, pp. 683–688, September 1974.

[23] V. Sidorenko and Z. Zyablov, "Decoding of Convolutional Codes Using a Syndrome Trellis," *IEEE Trans. Inform. Theory,* vol. IT–40, pp. 1663-1666, September 1994.

MSN Codes for Two-track, Two-head Magnetic Recording Systems

Erozan Kurtas John G. Proakis Masoud Salehi
Department of Electrical and Computer Engineering
Northeastern University, Boston, MA 02115

Abstract

In this article we outline the coding issues for multi-track magnetic recording systems. We provide analytical results related to uncoded and coded multi-track systems which constitute the basis of the code construction. We have designed multi-track MSN codes which give coding gains from 3dB to 6dB over uncoded systems.

1 Introduction

In single-track magnetic recording systems data is coded, modulated and written on tracks independently from neighboring or any other tracks, with the same modulation code used on each track. The main idea of a multi-track system is to encode, write and read in parallel. Decreasing the k constraint is always desirable in recording systems, because this brings better synchronization, consequently higher access speeds. However, the price of a low k constraint is a lower code-rate, in other words capacity loss. The advantage of using multi-track systems is the altered form of the k constraint. In a multi-track recording system, the k constraint on each channel is removed and a joint(vector) k constraint is imposed on the channel output sequences. As pointed out in [1], this provides higher rates and the information for timing and gain control can be obtained from any of the tracks which are coded jointly.

The multi-track system under investigation is modeled as a set of parallel partial-response channels. Each channel is of the form $(1-D)(1+D)^n$. In addition to this, the effect of the interference from each track to others is characterized by an inter-track interference (ITI) matrix, $\mathbf{T}$. Matched Spectral-Null (MSN) modulation codes which are intended for use on noisy partial-response channels with a finite input alphabet size were described in [2]. These codes provide a significant increase in the minimum Euclidean distance, limit the maximum run length of identical samples and are designed to eliminate quasi-catastrophic sequences. The simplest approach to designing MSN codes for multi-track systems is to code each track independently. However, independent coding of each track for a multi-track system is undesirable for two reasons. First, it does not have the effect of decreasing the k constraint. Second, such a scheme ignores the existence of ITI which is inherent in all multi-track systems under investigation and is not expected to provide adequate coding gains except in very special forms of inter-track interference.

The goal of any coding scheme employed in magnetic recording systems is to increase the areal density (density in bits per unit area). Increase in areal density in the radial direction can be achieved by narrowing the track width [3]. The drawback of such an approach is the loss in SNR due to track narrowing. To compensate for this loss one can employ coding on each track but this leads to a decrease in linear density. An overall gain in areal density can be obtained if high rate codes can be designed that can provide high coding gains to make up the loss in SNR.

As a result of track narrowing inter-track interference (ITI) becomes an important noise source in disc recording systems. Various researchers [4, 5, 6, 7] have shown that the effect of ITI can be reduced by means of multiple recording heads. Also as shown in [1] and [8], encoding, writing and reading data in parallel tracks jointly have advantages over conventional single-track systems. In a multi-track system the information for timing and gain control can be obtained from any track. This results from the altered form of the k-constraint. In multi-track systems k-constraint is described by the number of $\vec{0}$ code vectors, while each track must satisfy the d-constraint separately. A two track system is shown in Fig 1.

2 Multi-track Systems

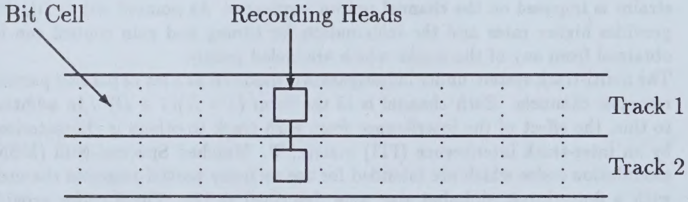


Figure 1: Two track recording system

All of the above remarks provide the motivation for designing joint coding schemes for multi-track applications. These codes have to combat two dimensional interference patterns, ISI in axial dimension and ITI in radial direction. The ability of these codes to compensate for the SNR loss incurred by track narrowing is yet to be investigated. Our primary research goal is to design multi-track codes which operate at high rates and give high enough gains not only to compensate for the SNR loss but to provide additional increase in areal density.

The effect of two dimensional interference patterns, intertrack and intersymbol, is described by $Z(D)$. We assume time wise independent and identically distributed but component wise correlated additive zero mean Gaussian noise. The symbols $\underline{\eta}(i)$ and $\underline{\eta}(i)$ are independent for $i \neq j$ but $[\underline{\eta}]_i(k)$ and $[\underline{\eta}]_j(k)$ are correlated with correlation matrix R. The schematic diagram for our model is shown in Fig 2.

$$\underline{y}(n) = \underline{x}(n) + \underline{\eta}(n)$$

$$\underline{X}(D) = Z(D)\underline{S}(D)$$

where $\underline{S}(D) = \sum_n \underline{s}(n)D^n$ and $Z(D) = [Z_{ij}(D)]$. $Z_{ij}(D) = j^{th}$ channel interference on the i^{th} channel. For example

$$Z_1(D) = \begin{bmatrix} 1-D & \alpha(1-D) \\ \alpha(1-D) & 1-D \end{bmatrix} \qquad (1)$$

for two $1-D$ channels interfering symmetrically with a coefficient α. This channel is illustrated in Fig 3.

It is obvious that the output of the channel is a function of the last N inputs, where N is the maximum length of interference, that is;

$$x(k) = f(k, \ldots, k-N+1)$$

It can be shown that the metric m for ML detection has the form

$$m = [\underline{f}(k, \ldots, k-N+1) - 2\underline{y}(k)]^t R^{-1} \underline{f}(k, \ldots, k-N+1)$$

For $Z_1(D)$, m reduces to

$$m = [\underline{f}(n, n-1) - 2\underline{y}(n)]^t R^{-1} \underline{f}(n, n-1)]$$

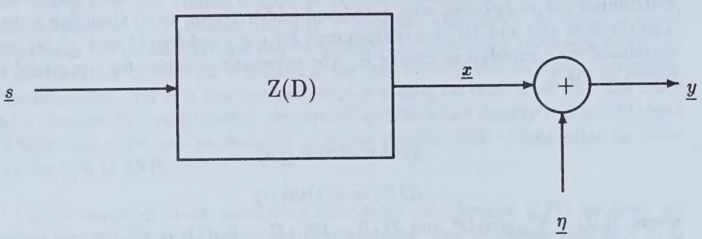
with the four state trellis shown in Fig 4. As an example, the metric for the dashed-line branch is

$$m = [\alpha - 1 - 2y_1 \;\; 1 - \alpha - 2y_2] R^{-1} [\alpha - 1 \;\; 1 - \alpha]^t$$

For the $R = I$ case we have

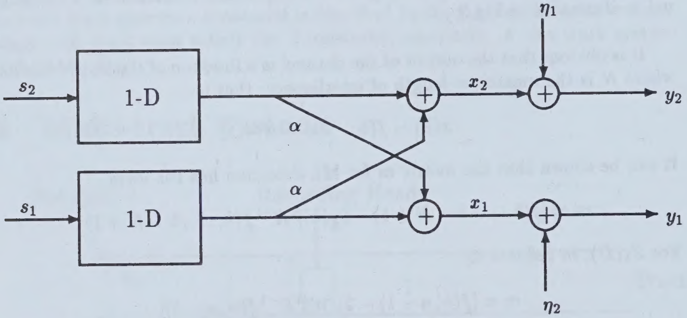
$$m = 2(\alpha - 1)^2 - 2(\alpha - 1)[y_2 - y_1]$$

We present the trellis corresponding to the Z_1 channel with $R = I$ in a tabular form in Table 1. The values for d^2_{min} of this trellis depend on the value of α. As we will see in the following section this is a crucial point for designing robust codes which can operate consistently under any value of α.



$\underline{s}$
Z(D)
$\underline{x}$
+
$\underline{y}$
$\underline{\eta}$

Figure 2: Model for 2 Dimensional systems



η_1
s_2
1-D
x_2
y_2
α
α
s_1
1-D
x_1
y_1
η_2

Figure 3: 2 track $1 - D$ channel with intertrack interference

	00	01	10	11
00	$0\ 0$	$\alpha\ 1$	$1\ \alpha$	$1+\alpha\quad 1+\alpha$
01	$-\alpha\ \ -1$	$0\ 0$	$1-\alpha\ \ \alpha-1$	$1\ \ \alpha$
10	$-1\ \ -\alpha$	$\alpha-1\ \ 1-\alpha$	00	$\alpha\ \ 1$
11	$-1-\alpha\ \ -1-\alpha$	$-1\ \ -\alpha$	$-\alpha\ \ -1$	$0\ 0$

Table 1: Trellis for Z_1 channel with the unmodified metric

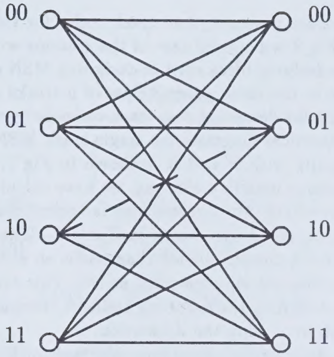


Figure 4: Trellis diagram for the 2 track system with intertrack interference.

	I	3	4	5	6	7	8	9	10
J									
3		0.7500	0.8002	0.8268	0.8427	0.8529	0.8598	0.8648	0.8685
4		0.8002	0.8471	0.8721	0.8870	0.8966	0.9031	0.9078	0.9113
5		0.8268	0.8721	0.8962	0.9106	0.9199	0.9263	0.9308	0.9342
6		0.8427	0.8870	0.9106	0.9248	0.9339	0.9401	0.9446	0.9478
7		0.8529	0.8966	0.9199	0.9339	0.9429	0.9490	0.9534	0.9567
8		0.8598	0.9031	0.9263	0.9401	0.9490	0.9551	0.9595	0.9627
9		0.8648	0.9078	0.9308	0.9446	0.9534	0.9595	0.9638	0.9670
10		0.8685	0.9113	0.9342	0.9478	0.9567	0.9627	0.9670	0.9702

Table 2: Capacities for various values of I and J

3 MSN codes for Two-Track Two-Head Recording Systems

In this section we consider designing two-track codes for the system shown in Fig 3. The system of Fig 3 is a special case of the systems with n-tracks and m-heads. However the underlying ideas used in designing MSN codes for two-track systems are applicable to the more general class of n-tracks and m-heads.

The beginning point for designing two-track codes is to consider the Cartesian product of two canonical diagrams for single track MSN codes as given in [2]. We denote this graph with **F** and it is shown in Fig 5. Using the largest eigenvalue of the adjacency matrix technique, we have calculated the capacities of various subdiagrams of the two dimensional canonical diagram. The resulting values for different I and J are shown in Table 2. The starting point for designing MSN codes for n parallel dicode channels is an n-dimensional canonical diagram which specifies the spectral nulls at DC. This raises the complexity issue, especially while designing the decoding trellises, compared to the capacity increases obtained with increasing the dimension.

Although one can design codes from any subgraph of **F**, it is important to have an analytical understanding of the code structure and be able to base the design issues on analytical terms. For this purpose, we investigate the distance properties of the uncoded system carefully and try to discover the major sources which cause performance degradation.

Formally, let $\mathbf{a} = \vec{a}_0, \ldots, \vec{a}_n$ and $\mathbf{b} = \vec{b}_0, \ldots, \vec{b}_n$ be sequences generated by paths in $\mathbf{F}$, $\Gamma_a = \{\sigma_0, \sigma_1^a, \sigma_2^a, \ldots, \sigma_n^a, \sigma_{n+1}^a\}$ and $\Gamma_b = \{\sigma_0, \sigma_1^b, \sigma_2^b, \ldots, \sigma_n^b, \sigma_{n+1}^b\}$. The sequence $\epsilon = \vec{\epsilon}_0, \ldots, \vec{\epsilon}_n$ with $\vec{\epsilon}_i = (a_{i1} - b_{i1}, a_{i2} - b_{i2})$ is referred as the difference sequence corresponding to **a** and **b** . If $\sigma_{n+1}^a = \sigma_{n+1}^b$ then ϵ is called a difference event, and if $\sigma_{n+1}^a = \sigma_{n+1}^b = \sigma_0$, then ϵ is called a difference cycle. The difference sequences corresponding to **a** and **b** on track 1 and track 2 are denoted as $\mathbf{e}_1$ and $\mathbf{e}_2$, respectively. . Let $\mathbf{A}_{(n+1)\times(n+1)}$ be the autocorrelation matrix corresponding to the $(1-D)$ channel and define an inner product in R^{n+1} as $< \mathbf{u}, \mathbf{v} >_{\mathbf{A}} = \mathbf{u}\mathbf{A}\mathbf{v}^t$ with $\mathbf{u}, \mathbf{v} \in R^{n+1}$.

For the network of two parallel $(1-D)$ channels with symmetric ITI coefficient α the distance between two allowable recorded sequences is given by [4], [9],

$$d^2(\epsilon) = (1+\alpha^2)(\| \mathbf{e}_1 \|_{\mathbf{A}}^2 + \| \mathbf{e}_2 \|_{\mathbf{A}}^2) + 2.2\alpha < \mathbf{e}_1, \mathbf{e}_2 >_{\mathbf{A}}.$$

The minimum distance corresponding to the uncoded system can be calculated using the trellis diagram shown in Table 1 or via the method of [9] and is given by the following proposition.

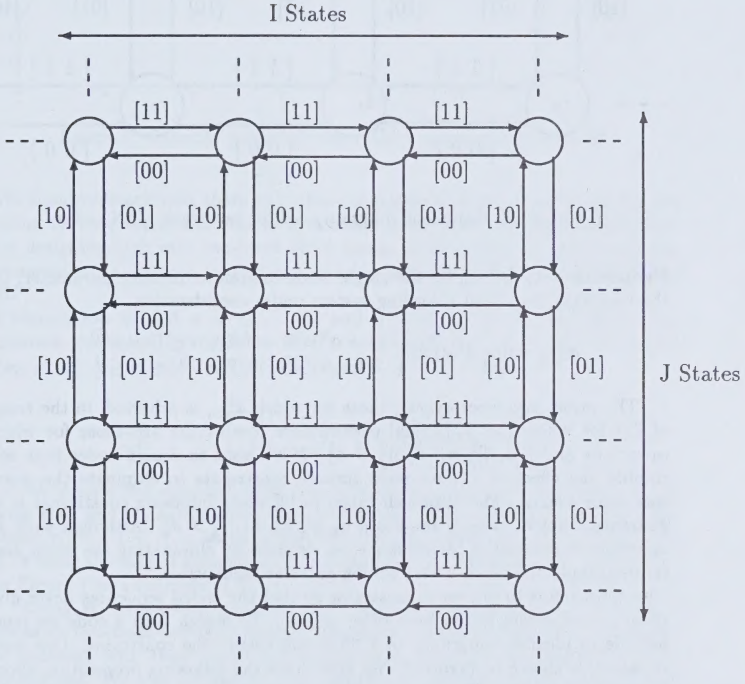

I States
J States
[11]
[00]
[10]
[01]

Figure 5: **F**, Canonical diagram for two dimensional DC-free sequences

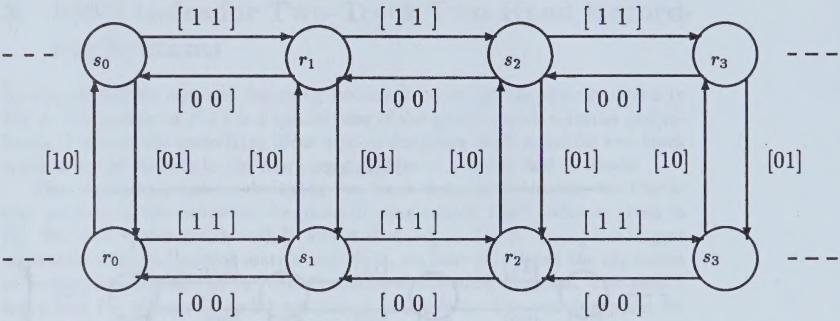


Figure 6: **H**, Canonical diagram used for two-track MSN codes

Proposition 1: Let d_0 be the single track minimum distance parameter, for the two-track, two-head recording system under consideration,

$$d^2_{min} = \min_{\epsilon \neq 0} d^2(\epsilon) = \begin{cases} (1+\alpha^2)d^2 0 & \text{if } 0 \leq \alpha \leq (2-\sqrt{3}) \\ 2(1-\alpha)^2 d_0^2 & \text{if } (2-\sqrt{3}) \leq \alpha \leq \frac{1}{2} \end{cases}$$

The worst case error events, those for which d^2_{min} is achieved, in the range of ITI for which the additional performance loss occurs are those for which $\mathbf{e}_1 = -\mathbf{e}_2$ and $\| \mathbf{e}_1 \|^2_{\mathbf{A}} = \| \mathbf{e}_2 \|^2_{\mathbf{A}} = d_0^2$. If we want to design codes that will combat the effect of ITI we must impose constraints to eliminate the worst case error events. The approach taken in [9] when imposing constraints is to guarantee that $\mathbf{e}_1 = -\mathbf{e}_2$ implies $\| \mathbf{e}_1 \|^2_{\mathbf{A}} = \| \mathbf{e}_2 \|^2_{\mathbf{A}} > d_0^2$. Although such an approach improves the performance, as we show by eliminating the worst case errors completely, much better coding gains are achievable.

Our approach is to impose a constraint so that the coded sequences never give rise to error events in the form of $\mathbf{e}_1 = -\mathbf{e}_2$. To design such a code we must be able to identify subgraphs of **F** that will satisfy the constraint. One such subgraph is shown in Figure 6. We then have the following proposition about **H** which shows that **H** indeed satisfies our constraint, hence it is suitable for code construction.

Proposition 2: Let **H** be the irreducible FSTD shown in Figure 6 and let **A** be the autocorrelation matrix of $the(1-D)$ channel. Let $\mathbf{a} = \vec{a}_0, \ldots, \vec{a}_n$ and $\mathbf{b} = \vec{b}_0, \ldots, \vec{b}_n$ be two distinct sequences with $\vec{a}_0 \neq \vec{b}_0$ generated by paths in **H**, $\Gamma_a = \{\sigma_0, \sigma_1^a, \sigma_2^a, \ldots, \sigma_n^a, \sigma_0\}$ and $\Gamma_b = \{\sigma_0, \sigma_1^b, \sigma_2^b, \ldots, \sigma_n^b, \sigma_0\}$. Let $\mathbf{e}_1$ and $\mathbf{e}_2$ be difference cycles corresponding to **a** and **b** on tracks 1 and 2, respectively. Then $\mathbf{e}_1 \neq -\mathbf{e}_2$ for all $\mathbf{e}_1, \mathbf{e}_2 \in \{0, 1, -1\}^{n+1}$.

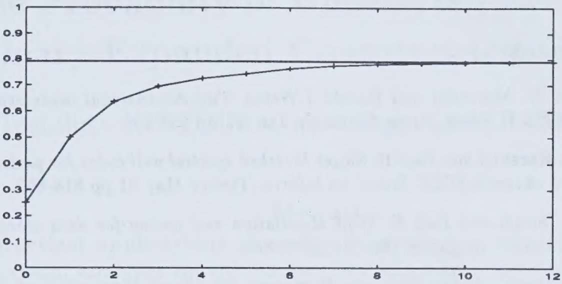


Figure 7: Capacity per track verses the number of states for $\mathbf{H}^r$

We should remark that there are other subgraphs of **F** that satisfy the Proposition 2. However **H** has various symmetry properties that make it attractive for designing high rate two-track MSN codes. In the following proposition we summarize the distance properties and achievable gains by codes designed from **H**.

Proposition 3: Let $\mathbf{a} = \vec{a}_0, \ldots, \vec{a}_n$ and $\mathbf{b} = \vec{b}_0, \ldots, \vec{b}_n$ be two distinct sequences generated by paths in **H**, $\Gamma_a = \{\sigma_0, \sigma_1^a, \sigma_2^a, \ldots, \sigma_n^a, \sigma_0\}$ and $\Gamma_b = \{\sigma_0, \sigma_1^b, \sigma_2^b, \ldots, \sigma_n^b, \sigma_0\}$, with $\vec{a}_0 \neq \vec{b}_0$, then

$$\min_{\epsilon \neq 0} d^2(\epsilon) = \begin{cases} 2(1+\alpha^2) & \text{if } 0 \leq \alpha \leq \frac{3-\sqrt{5}}{2} \\ 4(1-\alpha)^2 + 2\alpha & \text{if } \frac{3-\sqrt{5}}{2} \leq \alpha \leq \frac{1}{2} \end{cases}$$

We denote the subgraph of **H** which contains states r_i, s_i with $0 \leq i \leq N$ with $\mathbf{H}_N$, and the subgraph of the second power of **H** which contains only the states r_i, $0 \leq i \leq N$ is denoted $\mathbf{H}_N^r$. The capacity of $\mathbf{H}^r$ can be calculated as 0.79 using the largest eigenvalue of the adjacency matrix technique, and shown in Figure 7 as a function of N.

Using $\mathbf{H}^r$ as starting point we have designed rate 5/8, 2/3 and 3/4 MSN codes for two-track two-head system. The design process included the derivation of encoders with their associated sliding-block decoders using state-splitting techniques. We also derived reduced complexity detector trellis structures for all the codes from the second power of $\mathbf{H}^r$. Our simulation results agree with the claims of Proposition 3 and resulted in 3dB to 6dB coding gain over the uncoded systems when the intertrack interference coefficient (α) takes values in the range of [0, 0.5].

References

[1] Michael W. Marcellin and Harold J.Weber *Two-dimensional modulation codes* IEEE J. Select. Areas Commun. Jan. 92 pp 254-265.

[2] Razmik Karabed and Paul H. Siegel *Matched spectral null codes for partial-response channels* IEEE Trans. on Inform. Theory May 91 pp 818-855.

[3] Paul H. Siegel and Jack K. Wolf *Modulation and coding for data storage* IEEE commun. magazine Dec. 91 pp 68-86.

[4] L.C. Barbosa, *A Simultaneous Maximum Likelihood Estimation of Sequences for Interferring Channels*, IBM Res. Rep. RJ-5625.

[5] L.C. Barbosa, *Maximum Likelihood Sequence Estimators:A Geometric View*, IEEE Trans. Inform. Theory, vol. IT-31, no.2, pp.419-427, March 1989.

[6] L.C. Barbosa *Simultaneous Detection of Readback Signals from Interfering Magnetic Recording Tracks Using Array Heads*, IEEE Trans. Magn., vol. MAG-26, no. 5, pp. 2163-2165, Sept. 1990.

[7] W.L. Abbot, J.M. Cioffi, and H.K. Thapar, *Performance of Digital Magnetic Recording with equalization and Off-track Interference*, IEEE Trans. Magn., vol. MAG-27, no.1, pp.705-716, Jan. 1991.

[8] R.E. Swanson and J.K. Wolf, *A New Class of Two-dimensional RLL Recording Codes*, IEEE Trans. Magn., vol. MAG-28, no. 6, pp.3407-3416, Nov. 1992.

[9] Emina Soljanin and Costas N. Georghiades, "Coding for two-head recording Systems", IEEE Trans. on Inform. Theory, May 95 pp 747-755.

An Approximate Calculation Method of the Probability of Undetected Error for Binary Expanded Concatenated Codes

Toshihisa NISHIJIMA, and Shigeichi HIRASAWA

Abstract

In practical applications of coding theory, e.g., Signature analysis, ARQ system, analyses on the probability of undetected error for binary linear block codes play an important part. The analyses of this probability are also very interesting from the theoretical viewpoint. By utilizing the structure of concatenated codes constructed by concatenating the inner code whose code length is very small compared to that of the outer code and the outer code which is the maximum distance separable (MDS) code over GF(2^K), we proposed an approximate calculation method of the probability of undetected error for binary expanded concatenated codes used as error detecting codes.

I. INTRODUCTION

In practical applications of coding theory, e.g., Signature analysis, ARQ system, analyses on the probability of undetected error for binary linear block codes play an important part. The analyses on this probability is also very interesting from the theoretical viewpoint.

It is well known that the probability of undetected error, $P_u(\varepsilon)$, of an $(n,k,d)^\dagger$ code c is given by

$$P_u(\varepsilon) = \sum_{i=1}^{n} A_i \varepsilon^i (1-\varepsilon)^{n-i}, \tag{1}$$

where A_i is the number of codewords with Hamming weight i in the code c[1]. The exact value of the $P_u(\varepsilon)$ can be calculated by using the weight distribution of the code c. However, for almost all codes, the weight distribution is not explicitly known. Hence, some methods for calculating the upper bound on $P_u(\varepsilon)$ have been proposed[2]-[7]. It is also

---

T. Nishijima is with Department of Industrial and Systems Engineering, College of Engineering, Hosei University, 3-7-2, Kajinocho, Koganei-shi, Tokyo, 184 JAPAN. E-mail `nishi@nishi.is.hosei.ac.jp`

S. Hirasawa is with Department of Industrial Engineering and Management, School of Science and Engineering, Waseda University, 3-4-1, Ohkubo, Shinjuku-ku, Tokyo, 169 JAPAN.

The research leading to this paper is partially supported by the Ministry of Education under Grant-Aids 07558168 for Scientific Research.

[†] The (n,k,d) code denotes the code of length n, the number of information symbols k, and the minimum distance d. If d is obvious or is not necessary, d is omitted.

known by deriving the average probability of undetected error, $\overline{P}_u(\varepsilon)$, for the ensemble of all binary (n,k) linear block codes that there exist a binary (n,k) linear block codes for which the $P_u(\varepsilon)$ is upper bounded by $2^{-(n-k)}$[1]. Some explicitly constructive codes, e.g., Hamming codes, satisfying that $P_u(\varepsilon)<2^{-(n-k)}$ are found out for the finite code length n[2][3]. A code for which the $P_u(\varepsilon)$ is monotonically increasing in ε, for $0 \le \varepsilon \le \frac{1}{2}$ is called a proper code[3]. A proper code obviously bounded by $2^{-(n-k)}$ since $P_u(\frac{1}{2})= 2^{-(n-k)} - 2^{-n}$. It is shown that binary perfect codes are proper[3]. Furthermore, it is also shown that the codes not to reach to the Varshamov-Gilbert (V-G) bound are not proper[8]. That is, poor error correction codes are poor error detection codes. Of course, since the codes asymptotically meeting the V-G bound are not known, the proper codes are not explicitly shown for any code length n and code rate $r = \frac{k}{n}$, $0 < r < 1$.

On the other hand, a concatenation is a specific method of constructing long codes from shorter codes[9]. The constructive concatenated codes given by J. Justesen[10] are the first asymptotically good codes, thought these codes do not satisfy the V-G bound. It is shown that binary linear concatenated codes with Reed-Solomon (RS) codes as the outer code and varying nonsystematic inner code exist in this class which asymptotically meet the V-G bound[11]. However the probability of undetected error, $P_{Cu}(\varepsilon)$, of an (N_O, K_O, D_O) binary expanded concatenated codes C_O is not evaluated from both the practical and theoretical viewpoint. The probability of undetected error, $P_{Cu}(\varepsilon)$, of binary expanded concatenated codes C_O is given by

$$P_{Cu}(\varepsilon) = \sum_{i=1}^{N_O} A_{Ci}\varepsilon^i(1-\varepsilon)^{N_O-i}, \tag{2}$$

where A_{Ci} is the number of codewords with Hamming weight i in binary expanded concatenated codes C_O. Obviously, we can not obtain the weight distribution of the codes C_O.

Then, in this paper, by utilizing the structure of concatenated codes constructed by concatenating the inner code whose code length is very small compared to that of the outer code and the outer code which is the maximum distance separable (MDS) code over GF(2^K), we propose an approximate calculation method of the probability of undetected error of binary expanded concatenated codes.

Throughout this paper, we assume that the codes are the binary linear block codes, and channel, the binary symmetric channel, with cross-over probability ε. In general, we assume that $0 \le \varepsilon \le \frac{1}{2}$.

II. CONCATENATED CODES

A. Encoding Method

An (N_O, K_O, D_O) concatenated code C_O is formed from two codes : an (N, K, D) binary code C as the inner code and an (n, k, d) Reed-Solomon code c over GF(2^K) as the outer code.

Suppose that binary kK information symbols are fed into encoding system from the source. Treating successive k symbols as a 2^K-ary symbol, let the k information symbols encode into a codeword of an $(n,\ k,\ d)$ RS outer code over GF(2^K) by the outer encoder.

Next treating each of the 2^K-ary symbols of the $(n,\ k,\ d)$ RS code as binary symbols of length K, let each of the K information symbols encode into a codeword of an (N, K, D) inner code over GF(2) by the inner encoder, where the inner encoder generate n

codewords of the inner code C. The resulting code is an (N_O, K_O, D_O) binary expanded concatenated code C_O, where

$$\begin{aligned} N_O &= nN, & (3)\\ K_O &= kK, & (4)\\ D_O &= dD. & (5) \end{aligned}$$

B. Detecting Method

Detecting method by the code C_O is also done in two steps.

First, after partitioning received sequence of length N_O into n sequences of length N, the polynomial of partitioned each sequence is successively divided by the generator polynomial of the inner code C. If at least one of n remainder polynomials is not zero, an error is detected and we stop this detection procedure. If all of n remainder polynomials are zeros, the n information symbols of length K are fed to the detection procedure by the outer code c.

Next, these n K- digit bytes fed from the inner code detection procedure are detected according to the method by the outer code c. The polynomial of n K- digit bytes are divided by the generator polynomial of the RS outer code c over GF(2^K). If the remainder polynomial is not zero, an error is detected, and we stop this detection procedure. If it is zero, the received sequence of length N_O is regarded as a sequence equivalent to the transmitted codeword in the code C_O and is accepted by the receiver.

III. AN APPROXIMATE CALCULATION METHOD OF THE PROBABILITY OF UNDETECTED ERROR

A. An Approximate Calculation Method

We assume that the weight distribution $\{A_{Ii}\}$ of the inner code C where A_{Ii}, $i = 1, 2, \ldots, N$, is the number of codewords with Hamming weight i in the inner code C, is explicitly given. Since $N \ll N_O$, we can exhaustively compute the weight distribution $\{A_{Ii}\}$ of the inner code C.

Firstly, from the above assumption, we calculate the average weight and the average error probability per bit of undetected error patterns when the error detection procedure by the inner code C is over.

Let WD be the weight distribution $\{A_{Ii}\}$ without $A_{I0} = 1$ of the inner code C, and $|WD|$, the number of elements in the WD. Then, the total number of repeated choosing j elements from WD is $|WD|^j$. Also, let $H_j^{(k)} = \{A_{Iw_k(1)}, A_{Iw_k(2)}, \cdots, A_{Iw_k(j)}\}$, where k =1,2, ..., $|WD|^j$ be a certain pattern of these choices. Then, we can have the following two equalities :

$$|H_j^{(k)}| = \prod_{l=1}^{j} A_{Iw_k(l)}, \tag{6}$$

$$|W_j^{(k)}| = \sum_{l=1}^{j} w_k(l). \tag{7}$$

The $|H_j^{(k)}|$ is the total number of choosing j codewords from within each codeword with the weight $w_k(l)$, and the $|W_j^{(k)}|$, the total sum of weights of j codewords of this case.

We assume that all of $n = 2^K - 1$ remainder polynomials are zeros when the detection procedure by the inner code C is over and any j remainder polynomials among n remainder polynomials are faulty, i.e. the remainder polynomials are zeros due to the undetected error pattern. From the equalities (6) and (7), we have the following lemma with respect to the average weight and the average error probability per bit of undetected error sequences for the inner code detection.

Lemma 1 : *Let $\overline{w}$ be the average weight and p, the average error probability per bit of undetected error sequences when the error detection procedure by the inner code C is over. Then, we have*

$$\overline{w} = \sum_{j=1}^{n} \binom{n}{j} \sum_{k=1}^{|WD|^j} |H_j^{(k)}||W_j^{(k)}|\varepsilon^{|W_j^{(k)}|}(1-\varepsilon)^{N-|W_j^{(k)}|}, \tag{8}$$

and

$$p = \frac{1}{N_O}\overline{w}. \tag{9}$$

Secondly, we consider that the (n, k, d) maximum distance separable (MDS) code c which is used for error detection on discrete memoryless channel with $q = 2^K$ inputs and q outputs and assume that any symbol which is transmitted has the probability $1 - \epsilon$ of being received correctly and the probability $\frac{\epsilon}{q-1}$ of being changed into each of the $q - 1$ other symbols. Then, the probability of undetected error, $P_{MDSu}(\epsilon)$, for MDS code c is given by the following lemma[12].

Lemma 2 [12] : *The probability of undetected error, $P_{MDSu}(\epsilon)$, of the MDS code c is given by*

$$P_{MDSu}(\epsilon) = q^{-(n-k)}\{1 + \sum_{i=0}^{n-k-1} \binom{n}{i} (q^{n-k} - q^i)(\frac{\epsilon}{q-1})^i(1 - \frac{q\epsilon}{q-1})^{n-i} \\ -q^{n-k}(1-\epsilon)^n\}. \tag{10}$$

Finally, we give the value obtained from equation (10) for ϵ which satisfies $1 - \epsilon = (1 - p)^K$ from p calculated by the equalities (8) and (9), as the approximate value of the probability of undetected error, $P_{Cu}(\varepsilon)$, of binary expanded concatenated codes C_O. Then we have the following theorem.

Theorem 1 : *The approximate value of the probability of undetected error, $P_{Cu}(\varepsilon)$, of binary expanded concatenate codes C_O is given by*

$$P_{Cu}(\varepsilon) \approx q^{-(n-k)}\{1 + \sum_{i=0}^{n-k-1} \binom{n}{i} (q^{n-k} - q^i)(\frac{\epsilon}{q-1})^i(1 - \frac{q\epsilon}{q-1})^{n-i} \\ -q^{n-k}(1-\epsilon)^n\}, \tag{11}$$

where

$$1 - \epsilon = (1-p)^K, \tag{12}$$

and $q = 2^K$.

B. A Simple Example

We show the simple case to calculate the probability per bit of undetected sequence for the inner code detection.

We consider concatenated code which is constructed by concatenating the (8, 4, 4) extended Hamming code as the inner code C and (15, 13, 3) RS code c over GF(2^4) as the outer code. The weight distribution $\{A_{Ii}\}$ of the (8, 4, 4) extended Hamming code is given by $\{A_{I0} = 1,\ A_{I4} = 14,\ A_{I8} = 1\}$. Then, we have $WD = \{A_{I4} = 14,\ A_{I8} = 1\}$, and $|WD| = 2$. From the equalities (6) and (7), for example, for $j = 2$, we have

$$
\begin{aligned}
&|WD|^2 = 4,\\
&H_2^{(1)} = \{A_{Iw_1(1)}, A_{Iw_1(2)}\} = \{A_{I4}, A_{I4}\},\\
&H_2^{(2)} = \{A_{Iw_2(1)}, A_{Iw_2(2)}\} = \{A_{I4}, A_{I8}\},\\
&H_2^{(3)} = \{A_{Iw_3(1)}, A_{Iw_3(2)}\} = \{A_{I8}, A_{I4}\},\\
&H_2^{(4)} = \{A_{Iw_4(1)}, A_{Iw_4(2)}\} = \{A_{I8}, A_{I8}\},\\
&|H_2^{(1)}| = \prod_{l=1}^{2} A_{Iw_1(l)} = A_{I4}A_{I4} = 14 \times 14 = 196,\\
&|H_2^{(2)}| = \prod_{l=1}^{2} A_{Iw_2(l)} = A_{I4}A_{I8} = 14 \times 1 = 14,\\
&|H_2^{(3)}| = \prod_{l=1}^{2} A_{Iw_3(l)} = A_{I8}A_{I4} = 1 \times 14 = 14,\\
&|H_2^{(4)}| = \prod_{l=1}^{2} A_{Iw_4(l)} = A_{I8}A_{I8} = 1 \times 1 = 1,\\
&|W_2^{(1)}| = \sum_{l=1}^{2} w_1(l) = 4 + 4 = 8,\\
&|W_2^{(2)}| = \sum_{l=1}^{2} w_2(l) = 4 + 8 = 12,\\
&|W_2^{(3)}| = \sum_{l=1}^{2} w_3(l) = 8 + 4 = 12,\\
&|W_2^{(4)}| = \sum_{l=1}^{2} w_4(l) = 8 + 8 = 16.
\end{aligned}
$$

Similarly, for all j, $1 \leq j \leq 15$, we can calculate the $|H_j^{(k)}|$ and the $|W_j^{(k)}|$. Therefore, we can obtain the concrete value of the equalities (8) and (9) from these values.

Next, from the concrete value p obtained from the equality (9), we solve the equality (12) with respect to ϵ. Substituting this concrete value ϵ into the equality (11), we can calculate the approximate value of the probability of undetected error for the (120, 52, 12) binary expanded concatenated code C_O :

$$P_{Cu}(\varepsilon) \approx 16^{-2}\{1 + \sum_{i=0}^{1} \binom{15}{i} (16^2 - 16^i)(\frac{\epsilon}{15})^i (1 - \frac{16\epsilon}{15})^{15-i} - 16^2(1-\epsilon)^{15}\},$$

IV. DISCUSSION

By utilizing the structural characteristic of concatenated codes, we proposed a new approximate calculation method of the probability of undetected error for binary expanded concatenated codes used as error detecting codes. Then, we will try to show the effectiveness of the approximate calculation method by the computer simulation for some constructive concatenated codes. Furthermore, we will estimate the computational complexity to detect an error for the binary expanded concatenated codes.

The authors will present the paper discuss on the probability of undetected error and the computational complexity to detected an error for iterated codes[13]. Then, we will also try to compare the capability of expanded binary concatenated code with that of iterated codes.

ACKNOWLEDGEMENT

The Authors would like to thank Mr. K. Umezawa for his helpful assistance.

REFERENCES

[1] S. Lin and D. J. Costello, Jr., *Error Control Coding : Fundamentals and Applications* , Englewood Cliffs, NJ : Prentice-Hall, 1983.

[2] S. K. Leung-Yan-Cheong and M. Hellman, "Concerning a bound on undetected error probability," *IEEE Trans. Inform. Theory,* vol. IT-22, pp. 235-237, Mar. 1976.

[3] S. K. Leung-Yan-Cheong E. R. Barnes, and D. U. Friedman, "On some properties of the undetected error probability of linear codes," *IEEE Trans. Inform. Theory,* vol. IT-25, pp. 110-112, Jan. 1979.

[4] J. K. Wolf, A. M. Michelson, and A. H. Levesque, "On the probability of undetected error for linear block codes," *IEEE Trans. Commun.,* vol. COM-30, pp. 317-324, 1982.

[5] M. C. Lin, "Bounds on the undetected error probabilities of linear codes for both error correction and detection," *IEEE Trans. Inform. Theory,* vol. IT-36, pp. 1139-1141, Sep. 1990.

[6] T. Fujiwara, T. Kasami, and Shou-ping Feng, "On the monotonic property of the probability of undetected error for a shortened code," *IEEE Trans. Inform. Theory,* vol. IT-37, pp. 1409-1411, Sep. 1990.

[7] T. Klϕve, "Bounds on the worst case probability of undetected error," *IEEE Trans. Inform. Theory,* vol. IT-41, pp. 298-300, Jan. 1995.

[8] R. Padvani and J. K. Wolf,"Poor error correction codes are poor error detection codes," *IEEE Trans. Inform. Theory,* vol. IT-30, pp. 110-111, Jan. 1984.

[9] G. D. Forney, Jr., *Concatenate Codes* , Cambridge, MA : M. I. T., 1966.

[10] J. Justesen, "A class of constructive asymptotically good algebraic codes," *IEEE Trans. Inform. Theory,* vol. IT-18, pp. 652-656, Sept. 1972.

[11] C. Thommesen, "The existence of binary linear concatenated codes with Reed-Solomon outer codes which asymptotically meet the Gilbert-Varshamov bound," *IEEE Trans. Inform. Theory,* vol. IT-29, pp. 850-853, Nov. 1983.

[12] T. Kasami and S. Lin, "The probability of undetected error for the maximum distance separable codes," *IEEE Trans. Communications,* vol. COM-32, No. 9, pp. 998-1006, Sept. 1984.

[13] T. Nishijima and S. Hirasawa, "On the probability of undetected error and the computational complexity to detect an error for iterated codes," *in Proc. ISIT'95,* Whistler, 1995.

IDENTIFICATION OF LINEAR MULTIVARIABLE DYNAMICAL SYSTEMS WITH STATIC INPUT NONLINEARITIES

H. Al-Duwaish*, M. Nazmul Karim+, and V. Chandrasekar*
* Department of Electrical Engineering, Colorado State University,
Fort Collins, CO 80523.
+ Department of Chemical and Bioresource Engineering,
Colorado State University, Fort Collins, CO 80523.

Abstract

This paper introduces a new method for the identification of multiple input multiple output (MIMO) linear systems with static input nonlinearities. The proposed method uses a hybrid model of a multilayer feedforward neural network (MFNN) in cascade with an autoregressive moving average (ARMA) model. A recursive algorithm has been developed for the estimation of the parameters of the hybrid model. Simulation results show that the proposed identification method is very successful in estimating the static nonlinearities and the linear part of linear systems with static input nonlinearities.

1. Introduction

Multivariable dynamical systems with input static nonlinearities are very common in many industrial applications. In chemical processes, static nonlinearities appear as nonlinear valves, nonlinear actuators, and nonlinear sensors. In communication systems, static nonlinearities are used to model nonlinear filters, nonlinear channels and nonlinear amplifiers. Fig. 1 shows the structure of multivariable systems with input static nonlinearities. In general, the number of inputs is different from the number of outputs. The perturbation of any one input results in a response from more than one output. This is due to the internal coupling in the system. The signals between the static nonlinearities and the linear part are assumed to be inaccessible for measurement. The identification problem addressed here involves the identification of the linear part as well as the static nonlinearities at the inputs from the input-output data of the multivariable system. This approach preserves the structure of the system and provides valuable information for control applications.

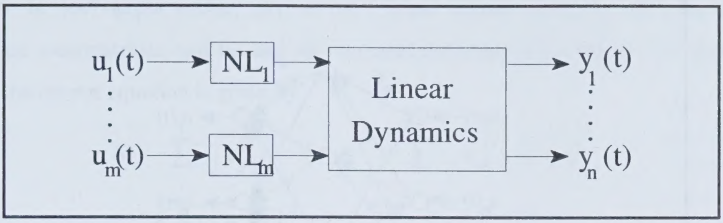


Fig. 1: Structure of multivariable systems with static input nonlinearities

In our previous study [1], a method for the identification of single input single output (SISO) systems with a static nonlinearity at the input was reported. In this paper, the identification method introduced in [1] is generalized to MIMO systems.

A large number of identification methods dealing with SISO and MISO systems can be found [2][3][4][5][6][7], however no effective method is currently available for MIMO linear systems with static input nonlinearities. This paper treats the identification of the general MIMO linear systems with static input nonlinearities. A simulated example is included to demonstrate the effectiveness of the proposed method.

2. MFNN/ARMA Identification Method

The structure of the MFNN/ARMA identification method for multivariable systems is shown in Fig. 2. The MFNN is used to model the static nonlinearities at the inputs and the ARMA model is used to model the linear part. The use of MFNN is motivated by their ability to model any nonlinear function to an arbitrary precision [8][9][10][11].

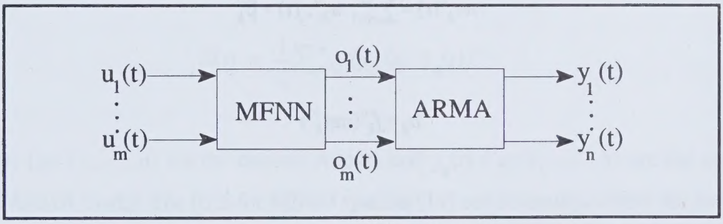


Fig. 2: Structure of the MFNN/ARMA model

Considering a nonlinear system with m inputs and n outputs, the MFNN will have m inputs and m outputs as shown in Fig. 3.

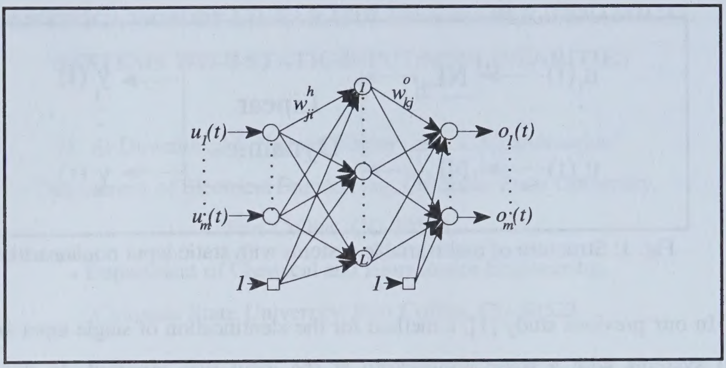


Fig. 3: Structure of the MFNN

An input vector , $U(t) = [u_1(t),........,u_m(t)]^T$, is applied to the input layer of the MFNN. The input to the jth hidden unit is given by

$$net_j^h(t) = \sum_{i=1}^m w_{ji}^h u_i(t) + \beta_j^h \tag{1}$$

where w_{ji}^h are the weights of the hidden layer and β_j^h is the bias term. The output of the jth hidden neuron is

$$z_j(t) = f_j^h(net_j^h(t)) \tag{2}$$

where f_j^h is the activation function. The equations for the output nodes are

$$net_k^o(t) = \sum_{j=1}^L w_{kj}^o z_j(t) + \beta_k^o \tag{3}$$

$$o_k = f_k^o(net_k^o) \tag{4}$$

where L is the number of neurons in the hidden layer, o_k denotes the kth output unit where $k = 1,...,m$. The output vector is given by $O(t) = [o_1(t),.........,o_m(t)]^T$. Now, the MFNN/ARMA model can be constructed by connecting the outputs of the MFNN to the ARMA model. The ARMA description of MIMO systems can be written as [14]

$$Y(t) = -A_1 Y(t-1) - A_2 Y(t-2) - .... - A_{na} Y(t-na) + B_0 O(t) + B_1 O(t-1) + ... + B_{nb} O(t-nb) \tag{5}$$

where $Y(t)$ is $n\times 1$ output vector, $O(t)$ is $m\times 1$ input vector, $A_1,....,A_{na}$ are $n\times n$ matrices, $B_0,....,B_{nb}$ are $n\times m$ matrices, and na and nb represent the order of the model. for a particular output q, the output equation is given by

$$\begin{aligned} y_q(t) = -\sum_{v=1}^{n} \overset{1}{a}_{qv} y_v(t-1) - .... - \sum_{v=1}^{n} \overset{na}{a}_{qv} y_v(t-na) \\ +\sum_{k=1}^{m} \overset{0}{b}_{qk} o_k(t) + ... +\sum_{k=1}^{m} \overset{nb}{b}_{qk} o_k(t-nb) \end{aligned} \tag{6}$$

Here the superscript associated with the a's and the b's denotes the matrix number, and the subscripts denote row and column numbers, respectively.

The identification method involves the estimation of the weights of the MFNN and the parameters of the linear ARMA model. This problem is solved by developing a new parameter estimation algorithm which is based on the well-known backpropagation (BP) and recursive least square (RLS) algorithms. This parameter estimation algorithm is called recursive least square with bakpropagation (RLS/BP).

3. RLS/BP Algorithm

The RLS algorithm is a standard procedure which has been derived to estimate the parameters of the linear ARMA models [14]. Also, the BP is used for estimating the weights of the MFNN. To estimate the weights and the parameters of the MFNN/ARMA model, the combination of these two algorithms is considered. This can be accomplished by defining the following error measure at the output of the ARMA model

$$E(t) = \frac{1}{2}\sum_{q=1}^{n} (y_{dq}(t)-y_q(t))^2 \tag{7}$$

Where $y_{dq}(t)$ ($q=1,........,n$) are the desired outputs and $y_q(t)$ ($q=1,........,n$) are the outputs of the MFNN/ARMA model. The RLS for MIMO systems [14] can be used to adjust the parameters of the ARMA model in a direction which will minimize the error defined by Eqn. 7. Trying to use the BP to adjust the weights of the MFNN, a problem arises in determining a measure of the error at the outputs of the MFNN. Intuitively, the error at the outputs of the MFNN must be related to the known error at the output of the ARMA model. To derive the rules for updating the weights of the MFNN, one substitutes Eqn. 6 into Eqn. 7 giving

$$E(t) = \frac{1}{2}\sum_{q=1}^{n} (y_{dq}(t) + \sum_{v=1}^{n} a_{qv}^{1} y_v(t-1) + \ldots\ldots + \sum_{v=1}^{n} a_{qv}^{na} y_v(t-na)$$
$$- \sum_{k=1}^{m} b_{qk}^{0} o_k(t) - \ldots\ldots - \sum_{k=1}^{m} b_{qk}^{nb} o_k(t-nb))^2$$

The *k*th input to the ARMA model, $o_k(t)$, at time *t* is equal to the *k*th output of the MFNN given by Eqns. 3 and. 4. Using linear activation functions for the output nodes of the MFNN and substituting Eqns. 3 and 4 into Eqn. 8, one gets

$$E(t) = \frac{1}{2}\sum_{q=1}^{n} (y_{dq}(t) + \sum_{v=1}^{n} a_{qv}^{1} y_v(t-1) + \ldots\ldots + \sum_{v=1}^{n} a_{qv}^{na} y_v(t-na)$$
$$- \sum_{k=1}^{m} b_{qk}^{0} (\sum_{j=1}^{L} w_{kj}^{o} z_j(t) + \beta_k^{o}) - \ldots\ldots - \sum_{k=1}^{m} b_{qk}^{nb} o_k(t-nb))^2$$

Now, to determine the direction in which to change the weights and biases of the MFNN, one calculates the gradient of $E(t)$. The weights can then be updated in the negative direction of the gradient. The gradient is

$$\nabla E(t) = \frac{\partial E(t)}{\partial w_{kj}^{o}} = (y_{dq}(t) - y_q(t)) \frac{\partial (y_{dq}(t) - y_q(t))}{\partial w_{kj}^{o}} \tag{11}$$

where

$$\frac{\partial (y_{dq}(t) - y_q(t))}{\partial w_{kj}^{o}} = - \sum_{k=1}^{m} b_{qk}^{0} z_j \tag{12}$$

Thus, the weights w_{kj}^{o} will be updated according to the following equation

$$w_{kj}^{o}(t+1) = w_{kj}^{o}(t) + \eta \sum_{q=1}^{n} (y_{dq}(t) - y_q(t)) \sum_{k=1}^{m} b_{qk}^{0} z_j \tag{13}$$

and the updating equation for the biases is

$$\beta_k^o(t+1) = \beta_k^o(t) + \eta \sum_{q=1}^{n} (y_{dq}(t) - y_q(t)) \sum_{k=1}^{m} b_{qk}^{0} \tag{14}$$

where η is the learning rate. Having found the updating equations for the weights and biases of the output layer, backpropagation learning algorithm can now be used to find the equations for updating the weights and biases of the hidden units which are given by

$$w_{ji}^{h}(t+1) = w_{ji}^{h}(t) + \delta_j^h u_i(t) \tag{15}$$

$$\beta_j^h(t+1) = \beta_j^h(t) + \delta_j^h \tag{16}$$

where

$$\delta_j^h = \acute{f}_j^{h}(net_j^{h}) \sum_{k=1}^{m} w_{kj}^{o} \left(\sum_{q=1}^{n} (y_{dq}(t) - y_q(t)) \sum_{k=1}^{m} b_{qk}^{0}\right) \tag{17}$$

and $\acute{f}_j^{h}$ is the derivative of the activation function of the hidden neurons.

4. Selection of Input Signals

Since the identification method involves the identification of the input nonlinearities, the input signals must be carefully selected to obtain consistent results. The input signals will be selected to satisfy the following requirements:

1) The input signals should span the entire operating ranges of the plant inputs.
2) The input signals should be uniformly distributed between the minimum and the maximum of the operating region of the nonlinearities.
3) The input signals should sufficiently rich in their frequencies.

The first and second requirements are needed to obtain reasonable approximation of the static nonlinearities in the operating region of the plant inputs. The third requirement is important in any identification process to assure parameter convergence [13]. Thus, uniformly distributed random variables will be used as inputs in the simulation example considered in this paper.

5. Simulation Results

Consider a process with two inputs and two outputs described by the following equations

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} 0.5 & -0.1 \\ 0.8 & -0.7 \end{bmatrix} \begin{bmatrix} y_1(t-1) \\ y_2(t-1) \end{bmatrix} + \begin{bmatrix} -0.3 & 0.2 \\ 0.9 & -0.5 \end{bmatrix} \begin{bmatrix} y_1(t-2) \\ y_2(t-2) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s_1(u_1(t)) \\ s_2(u_2(t) \end{bmatrix} \tag{18}$$

Where the static nonlinearities are given by

$$s_1(u_1(t)) = \frac{1 - e^{-4u_1(t)}}{1 + e^{-4u_1(t)}} \tag{19}$$

$$s_2(u_2(t)) = 0.5u_2^3(t) \tag{20}$$

Using random inputs uniformly distributed in the interval [-2,2], the desired outputs are generated using the process model. The proposed learning algorithm is used to update the weights of the MFNN and the parameters for the ARMA model. The matrices in the ARMA model converged to the following

$$A_1 = \begin{bmatrix} 0.5027 & -0.0951 \\ 0.7958 & -0.7015 \end{bmatrix}, \; A_2 = \begin{bmatrix} -0.3034 & 0.2044 \\ 0.8963 & -0.5018 \end{bmatrix}, \; B_0 = \begin{bmatrix} -0.3155 & 0.1846 \\ -0.0679 & 0.2464 \end{bmatrix} \tag{21}$$

It is clear that the matrices A_1 and A_2 are very close to the same matrices in the plant model. The matrix B_0 is different form the model, however, the multiplication of B_0 and the output of the MFNN converges to the nonlinearities at the inputs which are shown in Fig. 4 and Fig.5.

The results of this example and many other simulation examples reveal that the static nonlinearities and the linear part can be estimated accurately from input-output data using the proposed RLS/BP algorithm. The effects of noise, type of static nonlinearities, and linear dynamics on the convergence of the RLS/BP algorithm have been studied, and very promising results are obtained [15].

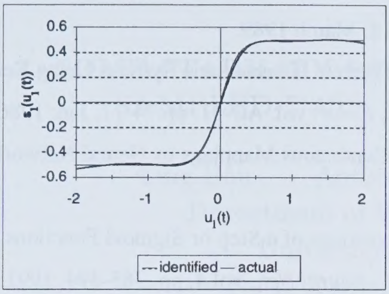


Fig. 4: Actual and identified nonlinearity at the first input

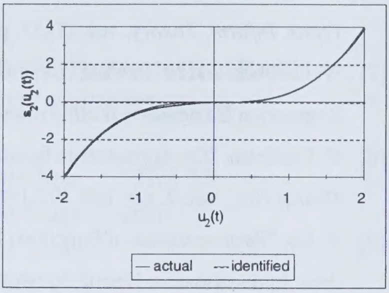


Fig. 5: Actual and identified nonlinearity at the second input

6. Conclusion

A new method for the identification of multivariable systems with static input nonlinearities has been developed. The new method consists of a MFNN in series with an ARMA model. A recursive identification algorithm has been derived for updating the weights of the MFNN and the parameters of the ARMA model. The new method is suitable for standard linear control design because the nonlinearities can be canceled by inserting the inverse of each nonlinearity in the appropriate location in the control loop. Exhaustive simulation studies suggest robust and satisfactory identification behavior of the developed method.

References

[1] H. Al-Duwaish, M. Nazmul Karim, and V. Chandrasekar, "A New Method for the Identification and Control of Hammerstein Model", *Proceedings of the Twenty-Ninth Conference on Information Science and Systems*, Johns Hopkins University, March 1995.

[2] E. Eskinat, S. H. Johnson, and W. L. Luyben,"Use of Hammerstein Models in Identification of Nonlinear Systems", *Amer. Ins. chem. Eng. J.*, vol. 37, pp. 255-268, 1991.

[3] P. G. Gallman, "A Comparison of Two Hammerstein Model Identification Algorithms", *IEEE Trans. Automat. Control*, vol. 21, pp. 124-126, 1976.

[4] K. S. Narenda and P. G. Gallman, "An Iterative Method for the Identification of Nonlinear Systems Using the Hammerstein Model", *IEEE Trans. Automat. Contr.*, vol. AC-11, pp. 546-550, July 1966.

[5] F. H. I. Chang and R. Luus, "A noniterative Method for Identification Using Hammerstein Model", *IEEE Trans. Automat. Contr.*, vol. AC-16, pp. 464-468, Oct. 1971.

[6] W. Greblicki and M. Pawlak, "Nonparametric Identification of Hammerstein Systems", *IEEE Trans. Inform. Theory*, vol. IT-35, pp. 409-418, March 1989.

[7] W. Greblicki and M. Pawlak, "Identification of Discrete Hammerstein Systems Using Kernel Regression Estimates", *IEEE Trans. Automat. Contr.* vol. AC-31, pp.74-77, Jan. 1986.

[8] K. Funahash, "On Approximate Realization of Continuous Mappings by Neural Networks", *Neural Net.*, vol. 2, pp. 183-192,1989.

[9] Y. Ito, "Representation of Functions by Superpositions of a Step or Sigmoid Functions and their Applications to Neurla Network Theory", *Neural Net.*, vol.4, pp. 385-394, 1991.

[10] K. Hornik, "Approximation Capabilities of Multilayer Feedforward Networks", *Neural Net.*, vol. 4, pp. 251-257, 1991.

[11] E. K. Blum and L. K. Li, "Approximation Theory and Feedforward Networks", *Neural Net.*, vol. 4, 511-515, 1991.

[12] J. A. Freeman and D. M. Skapura, *Neural Networks: Algorithms, Applications, and Programming Techniques*, Addison-Wesley, 1992.

[13] L. Ljung, *System Identification Theory for the User*, Prentice-Hall Inc.,1987.

[14] L. Ljung, and T. Soderstrom, *Theory and Practice of Recursive Identification*, The MIT press, 1987.

[15] H. Al-Duwaish, *Identification and Control of Nonlinear Dynamical Systems Using Multilayer Feedforward Neural Networks and Autoregressive Moving Average Models*, Ph. D. Dissertation, Colorado State University, 1995.

SOME STABILITY RESULTS FOR NONLINEAR SAMPLED-DATA CONTROL SYSTEMS *

Ling Hou Anthony N. Michel Hui Ye
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556, U.S.A.

Abstract

In the present paper we consider *nonlinear hybrid dynamical control systems* consisting of a *plant* (described by a system of *nonlinear ordinary differential equations*) and a *digital controller* (described by a system of *linear ordinary difference equations*). Associated with this nonlinear hybrid system is a *linear hybrid dynamical control system* consisting of the linearization of the plant (described by a system of linear ordinary differential equations) and the system of ordinary difference equations describing the digital controller. In the present paper we establish results which enable us to deduce the asymptotic stability and the instability of the null solution of the original nonlinear hybrid control system from its associated linearized hybrid control system.

1 Introduction

Because of significant advances in digital controller technology, the analysis and synthesis of sampled-data control systems are of continuing interest (see, e.g., [1]-[4] and references therein). In the present paper we consider sampled-data control systems which consist of an interconnection of the plant (described by a system of first order nonlinear ordinary differential equations) and a digital controller (described by a system of first order linear difference equations). The interfaces between the plant and the controller and between the controller and the plant are provided by A/D and D/A converters, respectively. The qualitative behavior of sampled-date control systems has been under investigation for many years, with an emphasis on linear systems (see, e.g., [1]-[4]). In the present paper, we show that under reasonable conditions the qualitative behavior of nonlinear sampled-data feedback systems can be deduced from the qualitative behavior of corresponding linearized sampled-data feedback systems. Although not entirely unexpected, we are not aware that this result has been reported previously.

The paper is organized as follows. In the next section the necessary notation is provided and in Section 3 we describe the class of systems considered in this paper. In Section 4 we establish a set of sufficient conditions for asymptotic stability and a set of sufficient condition for instability of an equilibrium for the class of systems considered herein.

Supported in part by the National Science Foundation under Grant ECS93-19352.

2 Notation

Let R denote the set of real numbers and let R^n denote real n-space. If $x \in R^n$, then $x^T = (x_1, \cdots, x_n)$ denotes the transpose of x. Let $R^{n\times m}$ denote the set of real $n \times m$ matrices. If $B = [b_{ij}]_{n\times m} \in R^{n\times m}$, then B^T denotes the transpose of B. A matrix $A \in R^{n\times n}$ is said to be *Schur stable* if all eigenvalues of A are within the unit circle of the complex plane. For $x \in R^n$, let $\|x\|$ denote the Euclidean vector norm, $\|x\| = (x^T x)^{1/2}$, and for $A \in R^{n\times n}$, let $\|A\|$ denote the norm of A induced by the Euclidean vector norm, i.e., $\|A\| = [\lambda_{max}(A^T A)]^{\frac{1}{2}}$.

Let R^+ denote the set of nonnegative real numbers, i.e., $R^+ = [0, +\infty)$, and let N denote the set of nonnegative integers, i.e., $N = \{0, 1, \cdots, \}$. For any $r \in R^+$, $[r]$ denotes the greatest integer which is less or equal to r.

Let X be a subset of R^n and let Y be a subset of R^m. We denote by $C[X, Y]$ the set of all continuous functions from X to Y, and we denote by $C^k[X, Y]$ the set of all functions from X to Y which have continuous derivatives up to and including order k.

3 A Class of Nonlinear Sampled-data Control Systems

In this paper we consider systems described by equation of the form

$$\begin{cases} \dot{x}(r) = f(x(r)) + Bu(k), & r \in [k, k+1) \\ u(k+1) = Cu(k) + Dx(k) & \end{cases} \tag{1}$$

where $x \in R^n$, $f \in C^1[R^n, R^n]$, $f(0) = 0$, $u \in R^m$, $B \in R^{n\times m}$, $C \in R^{m\times m}$, $D \in R^{m\times n}$, $r \in R^+$, and $k \in N$. We note that since $f(0) = 0$, $(x^T, u^T)^T = (0^T, 0^T)^T$ is an equilibrium of system (1).

In our considerations of the above sampled-data system, we did not include explicitly a description of the interface between the plant and the digital controller (a sample element) and between the digital controller and the plant (a sample and hold element).

To the authors' best knowledge, a systematic stability analysis of system (1) has not been conducted. In this paper we will show that the asymptotic stability of the equilibrium $(x^T, u^T)^T = (0^T, 0^T)^T$ of (1) can be deduced from the associated linear system given by

$$\begin{cases} \dot{x}(r) = Ax(r) + Bu(k), & r \in [k, k+1) \\ u(k+1) = Cu(k) + Dx(k), & k = 0, 1, 2, \cdots \end{cases} \tag{2}$$

where $A \in R^{n\times n}$ denotes the Jacobian of f evaluated at $x = 0$, i.e.,

$$A = [\frac{\partial f}{\partial x}(0)]_{n\times n}. \tag{3}$$

We note that in (1) the components of the state, x, u, are directly accessible as subsystem outputs (of the plant and the digital controller, respectively). When this is not the case, transducers are used to measure the states indirectly, resulting in linear output equations, as given for example in the system description

$$\left\{\begin{array}{ll} \dot{x}(r) = f(x(r)) + Bw(k), & r \in [k, k+1) \\ y(r) = Mx(r) + Nw(k), & r \in [k, k+1) \\ u(k+1) = Cu(k) + Dy(k), & k = 0, 1, \cdots \\ w(k) = Pu(k) + Qy(k), & k = 0, 1, \cdots \end{array}\right. , \tag{4}$$

where $y \in R^l$, $w \in R^s$, and M, N, P, and Q are real matrices of appropriate dimensions. By using the methodology employed herein, it is possible to establish a stability result of the type described above for system (1), using the linearization of system (4) about the equilibrium $(x^T, u^T)^T = (0^T, 0^T)^T$. We will not pursue this.

4 Asymptotic Stability and Instability Theorems

For the linear sampled-data system (2), we have the following result:

Lemma 1: The equilibrium $(x^T, u^T)^T = (0^T, 0^T)^T$ of the linear sampled-data system determined by the system of equations (2) is *uniformly asymptotically stable* if and only if the matrix

$$H \triangleq \begin{bmatrix} H_1 & H_2 \\ D & C \end{bmatrix}, \tag{5}$$

is Schur stable, where $H_1 = e^A$, $H_2 = \int_0^1 e^{A(1-\tau)} d\tau B$.

Proof: For the linear system (2) we can express $x(k+1)$ in terms of $x(k)$ and $u(k)$ explicitly as

$$\begin{aligned} x(k+1) &= e^A x(k) + \int_k^{k+1} e^{A(k+1-\tau)} Bu(k) d\tau \\ &= e^A x(k) + \int_0^1 e^{A(1-\tau)} d\tau Bu(k). \end{aligned} \tag{6}$$

Combining this with the second equation of (2) and letting $w(k) = [x(k)^T, u(k)^T]^T$, at these discrete points, the system can be viewed as a discrete-time linear system described by

$$w(k+1) = Hw(k).$$

When the matrix H is Schur stable, we know that $(x^T(k), u^T(k))^T$ converges to zero exponentially for $k \in N$. Furthermore, since

$$x(r) = e^{A(r-k)} x(k) + \int_k^r e^{A(r-\tau)} Bu(k) d\tau, \quad r \in [k, k+1)$$

it follows that

$$\|x(r)\| \leq e^{\|A\|} \|x(k)\| + \int_0^1 e^{\|A\|(1-\tau)} d\tau \|B\| \cdot \|u(k)\|$$

for all $r \in [k, k+1)$ and $k \in N$. Therefore $x(r)$ will converge to the origin, simultaneously with $u(k)$ and $x(k)$. □

Lemma 2: Assume that the matrix H, defined as in (5), has at least one eigenvalue outside the unit circle. Then the equilibrium $(x^T, u^T)^T = (0^T, 0^T)^T$ of the linear sampled-data system is *unstable.*

Proof: At the sampling instants, the system is defined by

$$w(k+1) = Hw(k).$$

This discrete-time system is unstable under the assumptions. Therefore the sampled-data system is unstable. □

The conclusion of Lemma 1 and Lemma 2 is well known (refer, e.g., to [1]- [4]). We are now in a position to prove the main result of the present paper.

Theorem 1: The equilibrium $(x^T, u^T)^T = (0^T, 0^T)^T$ of system (1) is *uniformly asymptotically stable* if the equilibrium $(x^T, u^T)^T = (0^T, 0^T)^T$ of the linear sampled-data system determined by the system of equations given in (2) is uniformly asymptotically stable, or equivalently, if the matrix H given in (5) is Schur stable.

Proof: Since $f \in C^1[R^n, R^n]$ and $f(0) = 0$, we can represent f as

$$f(x) = Ax + g(x), \tag{7}$$

where the matrix A is given in (3) and $g \in C^1[R^n, R^n]$ satisfies

$$\lim_{\|x\| \to 0} \frac{\|g(x)\|}{\|x\|} = 0 \ . \tag{8}$$

It follows from (8) that there exits a $\delta > 0$ such that

$$\|g(x)\| \leq \|x\| \tag{9}$$

whenever $\|x\| \leq \delta$. If we let

$$\delta_1 = \frac{e^{-(\|A\|+1)}}{\|B\| + 1}\delta,$$

then we can conclude that for any $k \in N$, it is true that $\|x(r)\| \leq \delta$ for all $r \in [k, k+1]$ whenever $\|x(k)\| \leq \delta_1$ and $\|u(k)\| \leq \delta_1$. For otherwise, there must exist an $r_0 \in (k, k+1)$ such that $\|x(r_0)\| = \delta$ and $\|x(r)\| \leq \delta$ for all $r \in [k, r_0]$. We will prove that this is impossible. Since for any $r \in [k, k+1]$ we have that

$$x(r) = x(k) + \int_k^r [Ax(\tau) + g(x(\tau)) + Bu(k)]d\tau, \tag{10}$$

it is true that

$$\|x(r)\| \leq \|x(k)\| + \int_k^r [\|A\| \cdot \|x(\tau)\| + \|g(x(\tau))\| + \|B\| \cdot \|u(k)\|]d\tau$$

for all $r \in [k, k+1]$. In particular, when $r \in [k, r_0]$, we have that

$$\begin{aligned} \|x(r)\| &\leq (\|x(k)\| + (r-k)\|B\| \cdot \|u(k)\|) + \int_k^r [\|A\| \cdot \|x(\tau)\| + \|g(x(\tau))\|]d\tau \\ &\leq (\|x(k)\| + \|B\| \cdot \|u(k)\|) + \int_k^r [\|A\| \cdot \|x(\tau)\| + \|x(\tau)\|]d\tau, \end{aligned} \tag{11}$$

where we have used in the last step of (11) the fact that $\|g(x(\tau))\| \le \|x(\tau)\|$, since $\|x(\tau)\| \le \delta$ for all $\tau \in [k, r_0]$, by assumption. By Gronwall's inequality (see, e.g., [9]), equation (11) implies that

$$\|x(r)\| \le (\|x(k)\| + \|B\| \cdot \|u(k)\|)e^{(\|A\|+1)(r-k)} \tag{12}$$

for all $r \in [k, r_0]$. Hence,

$$\begin{aligned} \|x(r_0)\| &\le (\|x(k)\| + \|B\| \cdot \|u(k)\|)e^{(\|A\|+1)(r_0-k)} \\ &\le (1+\|B\|)\delta_1 e^{(\|A\|+1)(r_0-k)} \\ &= (1+\|B\|)\frac{e^{-(\|A\|+1)}}{\|B\|+1}\delta \cdot e^{(\|A\|+1)(r_0-k)} \\ &= \delta \cdot e^{(\|A\|+1)(r_0-k-1)} \\ &< \delta \end{aligned} \tag{13}$$

since $r_0 < k+1$. Inequality (13) contradicts the assumption that $\|x(r_0)\| = \delta$. Therefore, we have shown that for any $k \in N$, it is true that $\|x(r)\| \le \delta$ for all $r \in [k, k+1]$ whenever $\|x(k)\| \le \delta_1$ and $\|u(k)\| \le \delta_1$. In view of (9), we can further conclude that

$$\|g(x(r))\| \le \|x(r)\| \tag{14}$$

for all $r \in [k, k+1]$ whenever $\|x(k)\| \le \delta_1$ and $\|u(k)\| \le \delta_1$. Inequality (14) implies that inequalities (11) and (12) hold for all $r \in [k, k+1]$, assuming that $\|x(k)\| \le \delta_1$ and $\|u(k)\| \le \delta_1$. Therefore, it follows from (12) that

$$\|x(r)\| \le (\|x(k)\| + \|u(k)\|) \cdot (1+\|B\|)e^{(\|A\|+1)} \tag{15}$$

for all $r \in [k, k+1]$, assuming that $\|x(k)\| \le \delta_1$ and $\|u(k)\| \le \delta_1$.

Since $f(x) = Ax + g(x)$, the first equation in (1) can be written as

$$\dot{x}(r) = Ax(r) + g(x(r)) + Bu(k), \tag{16}$$

for $r \in [k, k+1]$. The solution of equation (16) must have the form

$$x(r) = e^{A(r-k)}x(k) + \int_k^r e^{A(r-\tau)}Bu(k)d\tau + \int_k^r e^{A(r-\tau)}g(x(\tau))d\tau \tag{17}$$

for all $r \in [k, k+1]$. Specifically, when $r = k+1$, we have that

$$\begin{aligned} x(k+1) &= e^A x(k) + \int_k^{k+1} e^{A(k+1-\tau)}Bu(k)d\tau + \int_k^{k+1} e^{A(k+1-\tau)}g(x(\tau))d\tau \\ &= e^A x(k) + \int_0^1 e^{A(1-\tau)}d\tau \cdot Bu(k)d\tau + \Delta(k), \end{aligned} \tag{18}$$

where

$$\Delta(k) = \int_k^{k+1} e^{A(k+1-\tau)}g(x(\tau))d\tau. \tag{19}$$

We now proceed further by proving the following claim.

Claim 1: For any given $\epsilon > 0$, there exists a $\delta_3 > 0$, $\delta_3 \le \delta_1$, such that for any $k \in N$ it is true that

$$\|\Delta(k)\| < \epsilon(\|x(k)\| + \|u(k)\|)$$

whenever $\|x(k)\| < \delta_3$, and $\|u(k)\| < \delta_3$.

Proof: For the given $\epsilon > 0$, we choose $\epsilon_1 > 0$ such that

$$\epsilon = \epsilon_1 \cdot e^{(2\|A\|+1)}(1 + \|B\|).$$

We know by (8) that there must exist a $\delta_2 > 0$ such that

$$\|g(x)\| < \epsilon_1 \|x\| \tag{20}$$

whenever $\|x\| < \delta_2$. We choose

$$\delta_3 = min\Big\{\delta_1, \frac{1}{2\delta_2(1 + \|B\|)e^{(\|A\|+1)}}\Big\}.$$

Then, whenever $\|x(k)\| \leq \delta_3$ and $\|u(k)\| \leq \delta_3$, it is true by (12) that

$$\begin{aligned} \|x(r)\| &\leq (\|x(k)\| + \|u(k)\|) \cdot (1 + \|B\|)e^{(\|A\|+1)} \\ &\leq 2\delta_3(1 + \|B\|)e^{(\|A\|+1)} \\ &\leq \delta_2 \end{aligned} \tag{21}$$

for all $r \in [k, k+1]$. Combining (21), (20), and (15), we obtain that

$$\begin{aligned} \|g(x(r)\| &\leq \epsilon_1 \|x(r)\| \\ &\leq \epsilon_1(\|x(k) + \|u(k)\|) \cdot (1 + \|B\|)e^{(\|A\|+1)} \end{aligned} \tag{22}$$

for all $r \in [k, k+1]$ whenever $\|x(k)\| \leq \delta_3$, and $\|u(k)\| \leq \delta_3$. Hence, for $\Delta(k)$ given by (19), we know that

$$\begin{aligned} \|\Delta(k)\| &\leq \int_k^{k+1} e^{\|A\|} \cdot \|g(x(\tau))\| d\tau \\ &\leq e^{\|A\|} \cdot \epsilon_1(\|x(k)\| + \|u(k)\|) \cdot (1 + \|B\|)e^{(\|A\|+1)} \\ &= \epsilon \cdot (\|x(k)\| + \|u(k)\|) \end{aligned} \tag{23}$$

whenever the conditions $\|x(k)\| \leq \delta_3$, and $\|u(k)\| \leq \delta_3$ are satisfied, concluding the proof of *Claim 1*.

We are now in a position to prove the present theorem. Since by assumption H is Schur stable, where H is given in (5), we know that there exists a positive definite matrix P such that $H^T P H - P = -2I$, where $I \in R^{(n+m)\times(n+m)}$ denotes the identity matrix. We now define a Lyapunov function as

$$V(w) = w^T P w. \tag{24}$$

where $w \in R^{n+m}$. Let $w(k) = [x(k)^T, u(k)^T]^T$. For any $k \in N$, we have by (18) and (1) that

$$w(k+1) = Hw(k) + m(k), \tag{25}$$

where H is given by (5), $m(k)$ is given by

$$m(k) \triangleq (\Delta(k)^T, 0^T)^T, \tag{26}$$

and $\Delta(k)$ is given by (19). It is then true that

$$\begin{aligned} & V(w(k+1)) - V(w(k)) = w^T(k+1)Pw(k+1) - w^T(k)Pw(k) \\ = \;& [Hw(k)+m(k)]^T P[Hw(k)+m(k)] - w^T(k)Pw(k) \\ = \;& w^T(k)[H^TPH - P]w(k) + 2m^T(k)PHw(k) + m^T(k)Pm(k) \\ = \;& -2\|w(k)\|^2 + 2m^T(k)PHw(k) + m^T(k)Pm(k) \\ \leq \;& -2\|w(k)\|^2 + 2\|m(k)\| \cdot \|PH\| \cdot \|w(k)\| + \|m(k)\|^2 \cdot \|P\|. \end{aligned} \tag{27}$$

By the definition of $m(k)$, we know that $\|m(k)\| = \|\Delta(k)\|$. Furthermore, by *Claim 1*, if we choose an $\epsilon_0 > 0$ such that

$$2\sqrt{2}\epsilon_0\|PH\| + 2\epsilon_0^2\|P\| < 2, \tag{28}$$

then there exists a $\delta_3(\epsilon_0)$ such that

$$\|\Delta(k)\| < \epsilon_0(\|x(k)\| + \|u(k)\|), \tag{29}$$

whenever $\|x(k)\| < \delta_3(\epsilon_0)$, and $\|u(k)\| < \delta_3(\epsilon_0)$. Therefore, whenever $\|w(k)\| < \delta_3(\epsilon_0)$ (noticing that $\|w(k)\|^2 = \|u(k)\|^2 + \|x(k)\|^2$), it is true that

$$\|m(k)\| = \|\Delta(k)\| < \epsilon_0(\|x(k)\| + \|u(k)\|) \leq \epsilon_0\sqrt{2}\|w(k)\| \tag{30}$$

must hold. Combining (28) and (30), we conclude that

$$\begin{aligned} & V(w(k+1)) - V(w(k)) \\ < \;& -2\|w(k)\|^2 + 2\sqrt{2}\epsilon_0\|PH\| \cdot \|w(k)\|^2 + 2\epsilon_0^2\|P\| \cdot \|w(k)\|^2 \\ = \;& \|w(k)\|^2(-2 + 2\sqrt{2}\epsilon_0\|PH\| + 2\epsilon_0^2 \cdot \|P\|), \end{aligned} \tag{31}$$

whenever $\|w(k)\| < \delta_3(\epsilon_0)$.

Before concluding the proof, we require another intermediate result.

Claim 2: For any $k_0 \in N$, the inequality (31) holds for all $k \geq k_0$ whenever

$$\|w(k_0)\| < (\frac{\lambda_{min}(P)}{\lambda_{max}(P)})^{\frac{1}{2}}\delta_3(\epsilon_0), \tag{32}$$

where $\lambda_{min}(P)$ and $\lambda_{max}(P)$ denote the minimum and maximum eigenvalues of P, respectively.

Proof: Inequality (32) implies that

$$V(w(k_0)) \leq \lambda_{max}(P)\|w(k_0)\|^2 < \lambda_{min}(P)(\delta_3(\epsilon_0))^2.$$

Since $\|w(k_0)\| < \delta_3(\epsilon_0)$ is satisfied, we know by inequality (31) that $V(w(k_0+1))$ is less than $V(w(k_0))$. Therefore,

$$V(w(k_0+1)) < \lambda_{min}(P)(\delta_3(\epsilon_0))^2 \tag{33}$$

must be satisfied as well. Furthermore, since inequality (33) implies that $\|w(k_0+1)\| < \delta_3(\epsilon_0)$, it follows that $V(w(k_0+2))$ is less than $V(w(k_0+1))$, and $\|w(k_0+2)\|$ is less than $\delta_3(\epsilon_0)$. By induction, it follows that $\|w(k)\| < \delta_3(\epsilon_0)$ for all $k \geq k_0$. Hence (31) is satisfied for all $k \geq k_0$ as long as (32) is true. This concludes the proof of *Claim 2*.

By *Claim 2* $w(k)$ converges to zero exponentially, as long as (32) is true. Furthermore, noticing that $\|w(k)\| < \delta_3(\epsilon_0) < \delta_1$ whenever (32) is satisfied, it follows from (15) that $x(r)$ will converges to the origin, simultaneously with $x(k)$ and $u(k)$.

Noting that $\delta_3(\epsilon_0)$ is independent of k_0, we conclude that the equilibrium $(x^T, u^T)^T = (0^T, 0^T)^T$ is uniformly asymptotically stable if the matrix H (given by (5)) is Schur stable. This concludes the proof of the theorem. □

Theorem 2: Assume that the matrix H given in (5) has at least one eigenvalue outside the unit circle. Then the equilibrium $(x^T, u^T)^T = (0^T, 0^T)^T$ of the nonlinear sampled-data system is *unstable*.

Sketch of Proof: Similar to the proof of Theorem 1. By assumption, there exists a symmetric matrix P such that $H^T P H - P = -2I$, where $I \in R^{(n+m)\times(n+m)}$ denotes the identity matrix, and the Lyapunov function V is defined as previously. Notice that here P is not positive definite. It has at least one negative eigenvalue. Hence there exist points in every neighborhood of the origin where V is negative. On the other hand, by a similar argument as before, there exists a δ such that $V(w(k+1), k+1) - V(w(k), k) < 0$ whenever $\|w(k)\| < \delta$. Thus, $\|w(k)\|$ must exceed δ when $\|w(0)\| < \delta$ and $V(w(0)) < 0$. Therefore, the system is unstable. □

References

[1] P.A. Iglesias. " On the stability of sampled-data linear time-varying feedback systems". *Proc. 33rd CDC*, pp. 219-224, Lake Buena Vista, FL, Dec. 1994.

[2] B.A. Francis and T.T. Georgiou, "Stability theory for linear time-invariant plants with periodic digital controllers", *IEEE Trans. Automat. Contr.*, vol. 33, No. 9, pp. 820-832, 1988.

[3] T. Hagiwara and M. Araki. "Design of a stable feedback controller based on the multirate sampling of the plant output", *IEEE Trans. Automat. Contr.*, vol. 33, No. 9, pp. 812-819, 1988.

[4] T. Chen and B.A. Francis. "Input-output stability of sampled-data systems". *IEEE Trans. Autom. Contr.*, vol 33(9), pp. 820-832, Sept., 1988.

[5] A.N. Michel and K. Wang. *Qualitative Theory of Dynamical Systems*, Marcel Dekker, New York, 1995.

[6] T. Yoshizawa. *Stability Theory by Lyapunov's Second Method*, Math. Soc. Japan, Tokyo, 1966.

[7] V.I. Zubov. *Methods of A. M. Lyapunov and Their Applications*, P. Noordhoff Ltd., Groningen, The Netherlands, 1964.

[8] W. Hahn. *Stability of Motion*, Springer-Verlag, Berlin and New York, 1967.

[9] R.K. Miller and A.N. Michel. *Ordinary Differential Equations*, Academic Press, New York, 1982.

On Selecting Code Generators for TURBO Codes.*

Jan Seghers
Institute for Signal and Information Processing
Swiss Federal Institute of Technology
Zurich, SWITZERLAND

Lance C. Perez
Division of Engineering
The University of Texas at San Antonio

Daniel J. Costello, Jr.
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46616

Abstract

The performance of TURBO codes is addressed by examining the distance spectrum. The 'error floor' is shown to be a consequence of the relatively low free distance of the code. It is also shown that the 'error floor' can be lowered by increasing the size the interleaver. The excellent performance of TURBO codes at low signal-to-noise ratios is explained in terms of the distance spectrum. The interleaver in the TURBO encoder is shown to reduce the number of low weight codewords through a process called 'spectral thinning'. This thinned distance spectrum results in the free distance asymptote being the dominant performance parameter for low and moderate signal-to-noise ratios. Finally, it is shown that for a fixed interleaver size, the performance of TURBO codes may be improved by using a primitive polynomial as the feedback polynomial in the encoder.

---

*This work was supported by NASA grant NAG 5–557.

1 Introduction

The discovery of TURBO codes and the near capacity performance reported by the discoverers [1] has stimulated significant research effort to fully understand this new coding scheme [2]-[6]. In this paper, the performance of TURBO codes is explained in terms of the code's distance spectrum. These results explain both the near capacity performance of the TURBO codes and the observed "error floor" for moderate and high signal-to-noise ratios (SNR's).

2 Performance Bounds

In order to make clear the distinction between TURBO codes and convolutional codes, it is useful to consider these codes as block codes. To this end, the inputs sequences are restricted to sequences of length N, where N corresponds to the size of the interleaver in the TURBO encoder. With finite length input sequences of length N, a $(2,1,\nu)$ convolutional code may be viewed as a block code with 2^N codewords of length $2(\nu+N)$.

The bit error rate (BER) performance of the convolutional code with maximum likelihood (ML) decoding is upper bounded by

$$P_b \leq \sum_{i=1}^{2^N} \frac{w_i}{N}\frac{1}{2} erfc\left(\sqrt{d_i \frac{RE_b}{N_0}}\right)$$

where w_i and d_i are the information weight and total Hamming weight, respectively, of the i^{th} codeword. Collecting codewords of the same total Hamming weight and defining the average information weight per codeword as

$$\tilde{w}_d = \frac{w_d}{N_d},$$

where w_d is the total information weight of all codewords of weight d and N_d is the number of codewords of weight d, yields

$$P_b \leq \sum_{d=d_{free}}^{\infty} \frac{N_d \tilde{w}_d}{N}\frac{}{2} erfc\left(\sqrt{d \frac{RE_b}{N_0}}\right).$$

If a convolutional code has N_d^0 codewords of weight d caused by an information sequence $x(D)$ whose first one occurs at time 0, then it also has a codeword of weight d caused by the information sequence $Dx(D)$, a codeword of weight d caused by the information sequence $D^2x(D)$, and so on. Thus, as the length of the information sequences increases we have

$$\lim_{N\to\infty} \frac{N_d}{N} = N_d^0$$

and

$$\lim_{N\to\infty} \tilde{w}_d = \frac{w_d}{N_d} = \frac{w_d^0}{N_d^0}$$

and the bound on the BER of a convolutional code with ML decoding becomes

$$P_b \leq \sum_{d=d_{free}}^{\infty} \frac{w_d^0}{2} erfc\left(\sqrt{d \frac{RE_b}{N_0}}\right),$$

which is the standard union bound for ML decoding. For this reason, most efforts to find good convolutional codes have focused on finding codes that maximize the free distance d_{free} and minimize the multiplicity N_d for a given rate and constraint length.

The performance of the TURBO code with ML decoding is also bounded by

$$P_b \leq \sum_{i=1}^{2^N} \frac{w_i}{N}\frac{1}{2} erfc\left(\sqrt{d_i \frac{RE_b}{N_0}}\right)$$

where w_i and d_i are the information weight and total Hamming weight, respectively, of the i^{th} codeword. However, in the TURBO encoder the pseudorandom interleaver maps the sequence $x(D)$ to $x'(D)$ and the sequence $Dx(D)$ to a sequence $x''(D)$ that is different from $Dx'(D)$ with very high probability. Thus, the input sequences $x(D)$ and $Dx(D)$ produce different codewords with different Hamming weights.

Collecting codewords of the same total Hamming weight, the bound on the BER for TURBO codes becomes

$$P_b \leq \sum_{d=d_{free}}^{\infty} \frac{N_d\, \tilde{w}_d}{N}\frac{}{2} erfc\left(\sqrt{d \frac{RE_b}{N_0}}\right).$$

For TURBO codes with pseudorandom interleavers, $N_d \tilde{w}_d$ is much less than N and

$$\lim_{N\to\infty} \frac{\tilde{w}_d N_d}{N} << 1,$$

where

$$\frac{\tilde{w}_d N_d}{N}$$

is called the *effective multiplicity* of codewords of weight d.

3 Asymptotic Performance

In this section, the performance of TURBO codes for high SNR's is addressed using the bounds developed in the previous section. In order to make the points more concrete, the performance of the TURBO code is compared to the performance of a maximum free distance (MFD) $(2,1,14)$ convolutional code using simulations and analytical results. Figure 1 shows simulation results for the $(2,1,14)$ code with soft-decision Viterbi decoding and the TURBO code of [1] with iterative decoding and $N = 65536$.

It is well known that for high SNR's, the performance of a convolutional code with ML decoding is determined by the *free distance asymptote.* That is, for high SNR's the first term of the union bound is dominant. For the MFD $(2,1,14)$ code the first term in the bound is given by

$$\frac{w^0_{free}}{2} erfc\left(\sqrt{d_{free} \frac{RE_b}{N_0}}\right),$$

where $d_{free} = 18$, $N^0_{free} = 18$, and $w^0_{free} = 137$. In Figure 2, simulation results for this code are shown along with the free distance asymptote and it is clear that as the SNR gets large the asymptote accurately predicts the performance of the code.

For the TURBO code, the first term in the union bound is given by

$$\frac{N_{free}}{N}\frac{\tilde{w}_{free}}{2} erfc\left(\sqrt{d_{free} \frac{RE_b}{N_0}}\right),$$

where N is the size of the interleaver. For $N = 65536$, is has been found [7] that the TURBO code has $N_{free} = 3$ and $\tilde{w} = 2$. (This may vary slightly depending on which pseudorandom interleaver is used.) In Figure 3, the free distance asymptote is plotted together with simulation results. Once again, the asymptote is seen to be a good estimate of the performance of the code for high SNR's. Thus, it can be concluded that the 'error floor' of TURBO codes is due to their relatively low free distance and the corresponding free distance asymptote.

It can be shown [7] that as the interleaver size N increases, the number of free distance codewords approaches a constant K_{free} that is much less than N. That is,

$$\lim_{N\to\infty} N_{free} = K_{free}.$$

(This is not true for rectangular interleavers!) Thus, the 'error floor' can be lowered by increasing the size of the interleaver. Simulation results have shown that this is indeed the case.

4 Performance at Low SNR's

Having demonstrated that TURBO codes have a relatively small free distance, their excellent performance at low SNR's may seem even more surprising. In this section, the performance of TURBO codes at low SNR's is explained by examining the code's distance spectrum.

Returning to the MFD (2, 1, 14) convolutional code and Figure 2, it is clear that for low and moderate SNR's there is a significant gap between the free distance asymptote and the simulated performance. That is, the real coding gain is less than the asymptotic coding gain. The (2, 1, 14) code has the following distance spectrum

d	N_d^0	w_d^0
18	33	187
20	136	1034
22	835	7857
24	4787	53994
26	27941	361762
28	162513	2374453
30	945570	15452996
32	5523544	99659236

where N_d^0 is the number of paths of weight d and w_d^0 is the total information weight. In Figure 4, the contribution of each term in the distance spectrum to the overall performance of the code is plotted and compared to simulation results. It is easily seen that for SNR's less than 2.5 dB the free distance asymptote is the dominant term in the performance. This is due to the rapid growth of the multiplicities in the distance spectrum which causes the high weight paths to become dominant for low and moderate SNR's.

In view of the analysis of the (2, 1, 14) code, it is reasonable to suggest that the excellent performance of the TURBO code at low and moderate SNR's is due to a relatively "thin" distance spectrum. That is, the TURBO code is able to follow the free distance asymptote at lower SNR's because the multiplicities of higher weight codewords are small enough that the free distance asymptote remains the dominant term in the bound. The distance spectrum of TURBO codes is the result of a process called "spectral thinning" in which the interleaver effectively moves many lower weight codewords to higher weight codewords. This theory is supported by simulation results and actual calculations of the distance spectrum of some simple TURBO codes [7].

In Figure 5, simulation results are shown along with the contributions to the union bound of the first five terms of the distance spectrum of the TURBO code. (The distance spectrum of the TURBO was calculated assuming only information sequences of weight two.) Note that the free distance asymptote remains the dominant term in the bound even for low SNR's. Thus, the excellent performance of TURBO codes at low SNR's is a result of their relatively thin distance spectrum which enables the code to follow the free distance asymptote for low and moderate SNR's.

5 Primitive TURBO Codes

For sufficiently large interleavers, it can be shown that the first several terms of the distance spectrum are determined solely by information sequences of weight 2 [7]. As the size of the interleaver increases, the number of terms in the distance spectrum determined solely by weight 2 information sequences increases. Consequently, for a given interleaver size, the performance of TURBO codes can be improved by choosing the feedback polynomial in the encoder such that weight 2 information sequences generate high weight codewords.

Using primitive polynomials as the feedback polynomial in the encoder causes weight 2 information sequences to generate high weight codewords, thereby increasing the free distance of the TURBO code for larger interleaver sizes and improving the performance at moderate to high SNR's. Figure 6 shows simulation results for two TURBO codes with an interleaver size of 400 and memory $\nu = 4$ encoders. As expected, the TURBO code with the primitive feedback polynomial performs better at moderate and high SNR's due to the increased free distance.

6 Conclusion

The excellent performance of TURBO codes may be explained in terms of the distance spectrum of the code. The 'error floor' observed in simulations of TURBO codes is a manifestation of the free distance asymptote. Since TURBO codes have relatively low free distances, the free distance asymptote dominates performance at moderate and low error rates. The 'error floor' may be lowered by increasing the size of the interleaver. The exceptional performance of TURBO codes at low SNR's is due to 'spectral thinning' and the resultant ability of the code to follow the free distance asymptote at moderate and low SNR's. Finally, for a fixed interleaver size, the performance can be improved by using primitive feedback polynomials in the encoder. This increases the free distance of the overall code and results in better performance at moderate to high SNR's.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," Proc. 1993 IEEE Int. Conf. on Comm., Geneva, Switzerland, pp. 1064–1070, 1993.

[2] S. Benedetto and G. Montorsi, "Performance evaluation of TURBO-codes," *Electronics Letters*, **Vol. 31**, No. 3, pg. 163, February 2, 1995.

[3] S. Benedetto and G. Montorsi, "Average performance of parallel concatenated block codes," *Electronics Letters*, **Vol. 31**, No. 3, pg. 156, February 2, 1995.

[4] S. Benedetto and G. Montorsi, *Unveiling TURBO-codes: some results on parallel concatenated coding schemes*, Dipartimento di Elettronica, Politecnico di Torino, January 27, 1995.

[5] J. D. Andersen, *The TURBO Coding Scheme*, Report IT-146, Technical University of Denmark, June 1994.

[6] P. Robertson, "Illuminating the Structure of Parallel Concatenated Recursive Systematic (TURBO) Codes", Proc. GLOBECOM '94, Vol. 3, pp. 1298–1303, San Francisco, California, November 1994.

[7] J. Seghers, *On the Free Distance of TURBO Codes and Related Product Codes*, Final Report, Diploma Project SS 1995, Number 6613, Swiss Federal Institute of Technology, Zurich, Switzerland, August 1995.

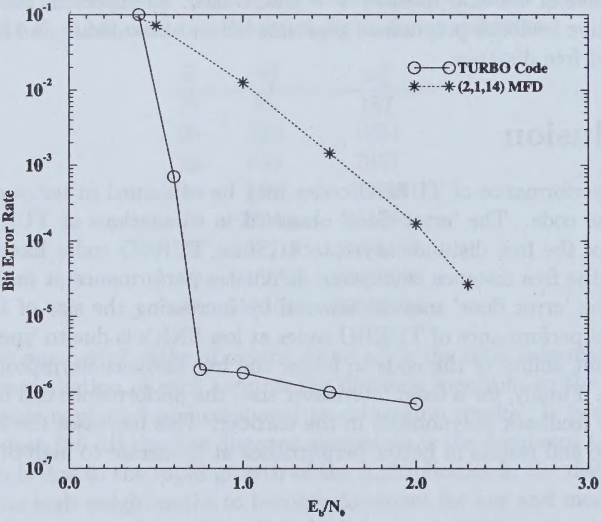


Figure 1: Simulation results for the TURBO code and a $(2, 1, 14)$ convolutional code.

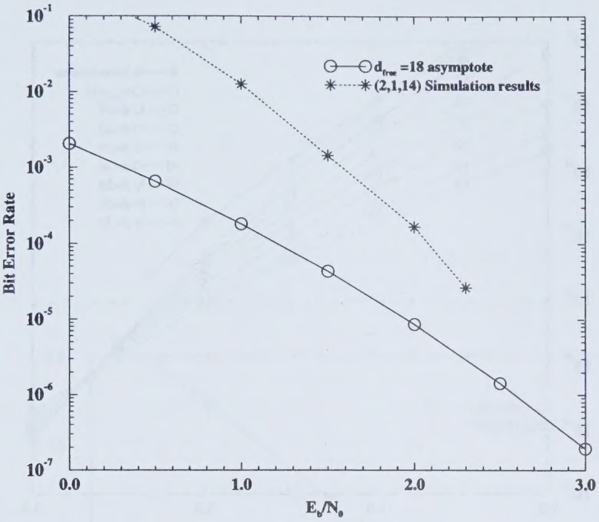


Figure 2: Asymptotic performance of the $(2, 1, 14)$ code.

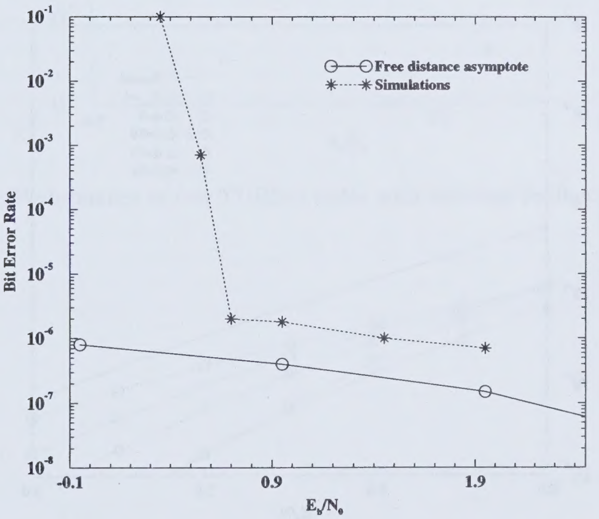


Figure 3: Asymptotic performance of the TURBO code.

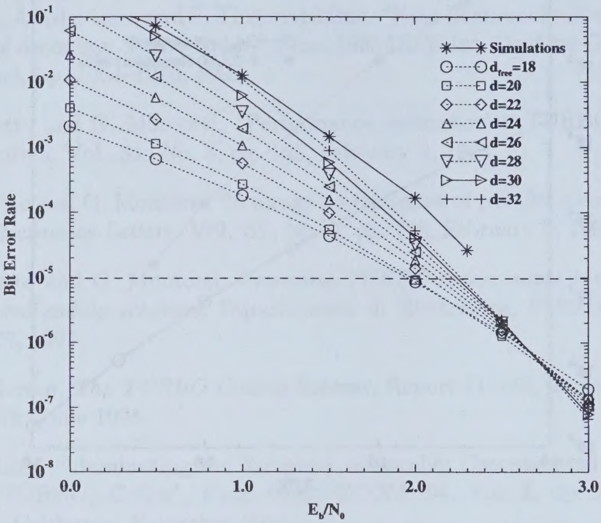


Figure 4: Decomposed performance of the MFD (2, 1, 14) code.

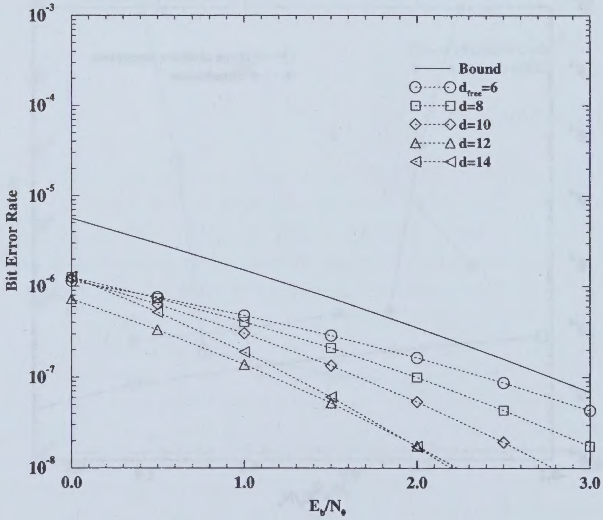


Figure 5: Decomposed performance of the TURBO code.

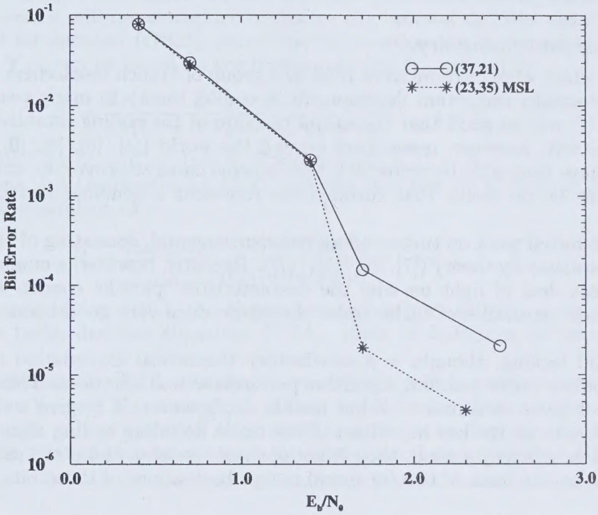


Figure 6: Performance of two TURBO codes with different feedback polynomials.

The Turbo Decision Algorithm*

Robert J. McEliece
Eugene R. Rodemich
Jung-Fu Cheng

Department of Electrical Engineering
California Institute of Technology
Pasadena, California 91125, USA

Abstract.

In this paper we will study the iterative decision algorithm that lies at the heart of the decoding algorithm for the recently discovered "turbo codes." We will show that, despite its excellent experimental behavior, the decision algorithm is suboptimal, not matter how many iterations are performed, and indeed may not even converge.

1. Introduction and Summary.

"Turbo codes," which were introduced in 1993 by a group of French researchers [7], are the most exciting and potentially important development in coding theory in many years. The claimed performance in [7] was so good that the initial reaction of the coding establishment was deep skepticism. Recently, however, researchers around the world ([4], [5], [8], [9], [10], [11], [12], [14], [15], [17]) have been able to reproduce, and in some cases improve, the announced results, so there can now be no doubt that turbo codes represent a genuine, and perhaps historic, breakthrough.

Much of the initial work on turbo-coding was experimental, consisting of simulation results without much explanatory theory ([7], [9], [15], [17]). Recently, however, a number of researchers have shed a great deal of light on why the characteristic "parallel concatenated systematic convolutional code" structure of turbo codes should produce very good codes ([4], [5], [6], [8], [12], [14], [16]).

What is still lacking, though, is a satisfactory theoretical explanation of why the low-complexity, iterative, turbo decoding algorithm performs as well as it does. This paper addresses this problem, but gives no solution. What it does do, however, is to give a clear formulation of what we believe to be the key ingredient of the turbo decoding coding algorithm: the turbo *decision* algorithm, whereby a good, though not optimal, decision about the value of a sequence of k bits is made on the basis of two (or more) noisy observations of those bits, via an iterative procedure.

We shall show that if $k = 1$ or $k = 2$, that the turbo decision is always optimal, but that for $k \geq 3$, the turbo decision need not be optimal, and indeed that the iterative procedure need not even converge.

We shall conclude by giving a simple example of a turbo code with $k = 3$, for which the turbo algorithm, despite its demonstrable shortcomings, nevertheless reaches the "right" decision with high probability. This phenomenon, which accounts for turbo codes having relatively low complexity, despite their high performance, is proposed as an important subject for further study.

---

* This work was partially supported by NSF grant no. NCR-9505975 and a grant from Pacific Bell.

2. Overview of the Turbo Algorithm.

Let $\mathbf{U} = (U_1, \ldots, U_k)$ be a k-dimensional random vector of 0's and 1's, whose components are independent, identically distributed random variables with common distribution

$$\Pr\{U_i = 0\} = \Pr\{U_i = 1\} = 1/2. \tag{2.1}$$

Informally, we think of $\mathbf{U}$ as an "information vector" whose components must be inferred from two "noisy versions of $\mathbf{U}$", $\mathbf{Y}_1$ and $\mathbf{Y}_2$, which are random n-vectors whose components lie in a set B. (The length n of the noisy vectors is in general not the same as the length of the information vector.) Conceptually, $\mathbf{Y}_i$ is obtained by passing $\mathbf{U}$ through a channel with transition probabilities $p_i(\mathbf{y}|\mathbf{u}) \overset{\text{def}}{=} \Pr\{\mathbf{Y}_i = \mathbf{y}|\mathbf{U} = \mathbf{u}\}$, for $i = 1, 2$. The channels are assumed to be independent in the sense that

$$\Pr\{\mathbf{Y}_1 = \mathbf{y}_1, \mathbf{Y}_2 = \mathbf{y}_2 | \mathbf{U} = u\} = p_1(\mathbf{y}_1|\mathbf{u})p_2(\mathbf{y}_2|\mathbf{u}). \tag{2.2}$$

The "turbo algorithm," to be described in detail below, is a suboptimal, iterative, method for making an intelligent decision about the value of each of the information bits $U_1, \ldots, U_k$, based on the two noisy observations $\mathbf{Y}_1$ and $\mathbf{Y}_2$. But before introducing the *suboptimal* turbo algorithm, however, it will be instructive to discuss the *optimal* decision algorithm.

An *optimal bit decision* (OBD), about the information bit U_i, based on an observation $(\mathbf{y}_1, \mathbf{y}_2)$ of $(\mathbf{Y}_1, \mathbf{Y}_2)$, can be based on the likelihood ratio R_i, defined by

$$R_i = \frac{\Pr\{U_i = 1 | \mathbf{Y}_1 = \mathbf{y}_1, \mathbf{Y}_2 = \mathbf{y}_2\}}{\Pr\{U_i = 0 | \mathbf{Y}_1 = \mathbf{y}_1, \mathbf{Y}_2 = \mathbf{y}_2\}} \qquad i = 1, \ldots, k. \tag{2.3}$$

If $R_i \geq 1$, the OBD for U_i is $U_i = 1$; if $R_i < 1$, the OBD is $U_i = 0$.

Using Lemma A.1 (Appendix A), (2.2), and the fact that $\Pr\{\mathbf{U} = \mathbf{u}\} = 2^{-k}$ for all $\mathbf{u}$, we find that (2.3) is equivalent to

$$R_i = \frac{\sum_{\mathbf{u}:u_i=1} p_1(\mathbf{y}_1|\mathbf{u})p_2(\mathbf{y}_2|\mathbf{u})}{\sum_{\mathbf{u}:u_i=0} p_1(\mathbf{y}_1|\mathbf{u})p_2(\mathbf{y}_2|\mathbf{u})} \qquad i = 1, \ldots, k. \tag{2.4}$$

It is thus seen that in the OBD, the channel transition probabilities p_1 and p_2 are tightly coupled. In the turbo decision algorithm (TDA), these probabilities are much more loosely coupled.

Roughly speaking, the TDA consists of two modules which compute likelihood ratios for the U_i's based on $\mathbf{Y}_1$ and $\mathbf{Y}_2$, separately. These two modules then iteratively communicate their results to each other, updating their likelihood ratios as they go, until a consensus is reached about the value of each information bit. The general structure is as shown in Figure 1.

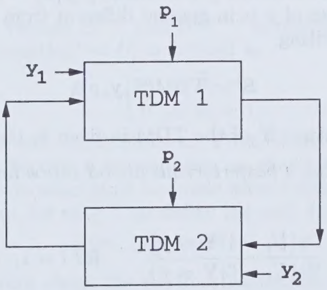


Figure 1. The Turbo decision algorithm. The mth estimate of the likelihood ratio vector $\mathbf{R}$ is $\widehat{\mathbf{R}}^{(m)} = \mathbf{S}^{(2m)}\mathbf{S}^{(2m+1)}$. The turbo decision modules TDM1 and TDM2 are identical devices, and are described in Section 2.

We shall describe the operation of the *turbo decision modules* TDM1 and TDM2 in the next section.

3. The Turbo Decision Module.

Conceptually, the TDM (see Figure 2) is a device which takes as input a k dimensional vector with positive components $\mathbf{S} = (S_1, \ldots, S_k)$, a vector $\mathbf{y} \in B^n$, and a set of conditional probabilities $p(\mathbf{y}|\mathbf{u})$ for all pairs $(\mathbf{y}, \mathbf{u}) \in B^n \times \{0,1\}^k$, and in response computes a modified vector $\mathbf{S}' = (S_1', \ldots, S_k')$, using the formula

$$S_i' = \frac{\sum_{\mathbf{u}:u_i=1} p(\mathbf{y}|\mathbf{u}) \prod_{j\neq i} S_j^{u_j}}{\sum_{\mathbf{u}:u_i=0} p(\mathbf{y}|\mathbf{u}) \prod_{j\neq i} S_j^{u_j}}, \qquad \text{for } i = 1, 2, \ldots, k. \tag{3.1}$$

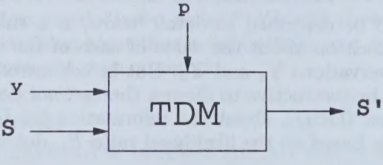


Figure 2. The Turbo decision module.

The components of $\mathbf{S}$ should be thought of as *a priori* likelihood ratios for the components of an unknown binary vector $\mathbf{U} = (U_1, \ldots, U_k)$ *whose components are assumed to be independent.* Thus if $p^0(\mathbf{u}) \overset{\text{def}}{=} \Pr\{\mathbf{U} = \mathbf{u}\}$, then we have

$$p^0(\mathbf{u}) = p^0(u_1) \cdots p^0(u_k), \tag{3.2}$$

where $p^0(u_i) \overset{\text{def}}{=} \Pr\{U_i = u_i\}$, and the *a priori* likelihoods ratios are

$$S_i = \frac{\Pr\{U_i = 1\}}{\Pr\{U_i = 0\}} \qquad \text{for } i = 1, \ldots, k. \tag{3.3}$$

The vector $\mathbf{y}$ should be thought of as a "noisy" version of $\mathbf{U}$, which has been transmitted over a channel defined by the conditional probabilities $p(\mathbf{y}|\mathbf{u}) = \Pr\{\mathbf{Y} = \mathbf{y}|\mathbf{U} = \mathbf{u}\}$ (note again that the number of components of $\mathbf{y}$ is in general different from the number of components of $\mathbf{u}$). We summarize (3.1) by writing

$$\mathbf{S}' = \mathrm{TDM}(\mathbf{S}, \mathbf{y}; p). \tag{3.4}$$

The significance of the output $\mathbf{S}'$ of the TDM is given in the following theorem.

3.1 Theorem. *If we denote the a posteriori likelihood ratios for the components of* $\mathbf{U}$ *by* R_i, *i.e.,*

$$R_i = \frac{\Pr\{U_i = 1|\mathbf{Y} = \mathbf{y}\}}{\Pr\{U_i = 0|\mathbf{Y} = \mathbf{y}\}} \qquad \textit{for } i = 1, \ldots, k, \tag{3.5}$$

then if S_i' *is as given in (3.1), we have*

$$R_i = S_i' \cdot S_i \qquad \textit{for } i = 1, \ldots, k. \tag{3.6}$$

Proof: From Lemma A.1 in Appendix A, we have

$$R_i = \frac{\sum_{\mathbf{u}:u_i=1} p(\mathbf{y}|\mathbf{u}) p^0(\mathbf{u})}{\sum_{\mathbf{u}:u_i=0} p(\mathbf{y}|\mathbf{u}) p^0(\mathbf{u})} \qquad \text{for } i = 1, 2, \ldots, k. \tag{3.7}$$

By (3.3), and the fact that $\Pr\{U_i = 0\} + \Pr\{U_i = 1\} = 1$, it follows that

$$\begin{aligned}\Pr\{U_i = 0\} &= 1/(1+S_i)\\ \Pr\{U_i = 1\} &= S_i/(1+S_i).\end{aligned} \tag{3.8}$$

The two cases in (3.8) can be combined into a single formula:

$$\Pr\{U_i = a\} = \frac{S_i^a}{1+S_i}, \quad \text{for } a = 0 \text{ or } 1, \tag{3.9}$$

and so, by (3.2), we have

$$p^0(\mathbf{u}) = \prod_{i=1}^{k} \frac{S_i^{u_i}}{1+S_i}. \tag{3.10}$$

Combining (3.10) and (3.7), we find that

$$\begin{aligned} R_i &= \frac{\sum_{u_i=1} p(\mathbf{y}|\mathbf{u}) \prod_{j=1}^{k} S_j^{u_j}}{\sum_{u_i=0} p(\mathbf{y}|\mathbf{u}) \prod_{j=1}^{k} S_j^{u_j}}\\ &= S_i \cdot \frac{\sum_{u_i=1} p(\mathbf{y}|\mathbf{u}) \prod_{j\neq i} S_j^{u_j}}{\sum_{u_i=0} p(\mathbf{y}|\mathbf{u}) \prod_{j\neq i} S_j^{u_j}}\\ &= S_i \cdot S_i'. \end{aligned}$$

Thus (3.6) follows. ▪

4. The Turbo Decision Algorithm.

Here is the formal definition of the turbo algorithm. Starting with observations $\mathbf{y}_1$ and $\mathbf{y}_2$, the computation loop shown in Figure 1 is initialized by feeding an "initial likelihood vector" $\mathbf{S}^{(0)} = (1, 1, \ldots, 1)$, into TDM 1. The successive outputs of the TDM's are then observed. The notation we shall use is as follows.

$$\begin{aligned} \mathbf{S}^{(0)} &= (1, 1, \ldots, 1) \quad \text{(initialization)}\\ \mathbf{S}^{(2m)} &= \mathrm{TDM}(\mathbf{S}^{(2m-1)}, \mathbf{y}_2) \quad \text{for } m \geq 1\\ \mathbf{S}^{(2m+1)} &= \mathrm{TDM}(\mathbf{S}^{(2m)}, \mathbf{y}_1) \quad \text{for } m \geq 0 \end{aligned} \tag{4.1}$$

In principle, the *turbo decision* is based on the limiting value of the product of consecutive outputs of the TDM's in the computation loop, i.e., on the *limiting turbo likelihood estimate*

$$\widetilde{\mathbf{R}} = \lim_{m\to\infty} \mathbf{S}^{(m)}\mathbf{S}^{(m-1)}. \tag{4.2}$$

(In (4.2), the product of the vectors $\mathbf{S}^{(m)}$ and $\mathbf{S}^{(m-1)}$ is defined componentwise.) Then the *turbo decision* for the ith information bit U_i is defined as

$$\widehat{U}_i = \begin{cases} 1 & \text{if } \widetilde{R}_i \geq 1\\ 0 & \text{if } \widetilde{R}_i < 1. \end{cases} \tag{4.3}$$

There are two difficulties with this definition. First, there is no guarantee that the limit in (4.2) exists, and indeed in Section 6 we will give an example for which the limit does not exist. Second, in practice, the decision must be made after a finite amount of computation. To circumvent both the difficulties, for $m \geq 1$ we define the mth *turbo likelihood estimate* as

$$\widetilde{\mathbf{R}}^{(m)} = \mathbf{S}^{(m)}\mathbf{S}^{(m-1)}. \tag{4.4}$$

Then the *mth turbo decision* about the ith information bit U_i is

$$\widehat{U}_i^{(m)} = \begin{cases} 1 & \text{if } \widetilde{R}_i^{(m)} \geq 1\\ 0 & \text{if } \widetilde{R}_i^{(m)} < 1 \end{cases} \tag{4.5}$$

Note that because of Theorem 3.1, each of the vectors in (4.4) is the *a posteriori* likelihood vector from one of the TDM's.

The question we shall address in the remainder of the paper is whether or not the turbo decision, defined by (4.3), and based on the limiting likelihood rations defined in (4.2), agrees with the OBD based on the true likelihood ratios defined in (2.3) or (2.4).

5. Algorithm Performance for $k = 1$ and $k = 2$.

The case $k = 1$ is trivial. For with $k = 1$, the OBD as given by (2.4) is

$$R_1 = \frac{p_1(\mathbf{y}_1|u = 1)p_2(\mathbf{y}_2|u = 1)}{p_1(\mathbf{y}_1|u = 0)p_2(\mathbf{y}_2|u = 0)}. \tag{5.1}$$

On the other hand, using (3.1) and (4.1), we have that

$$\begin{aligned} \mathbf{S}^{(0)} &= (1) \\ \mathbf{S}^{(2m)} &= p_2(\mathbf{y}_2|u = 1)/p_2(\mathbf{y}_2|u = 0) \quad \text{for all } m \geq 1 \\ \mathbf{S}^{(2m+1)} &= p_1(\mathbf{y}_1|u = 1)/p_1(\mathbf{y}_1|u = 0) \quad \text{for all } m \geq 0 \end{aligned}$$

so that (see (4.4)), $\widetilde{\mathbf{R}}^{(m)} = R_1$, where R_1 is defined in (5.1), for all $m \geq 2$. Thus for $k = 1$ the turbo decision algorithm converges in one iteration to the OBD.

For $k = 2$, we will show that the TDA always converges, and that while $\widetilde{\mathbf{R}}$ may not be the same as the optimal $\mathbf{R}$ defined in (2.3), nevertheless, the turbo *decision* is always the same as the OBD, i.e., R_i and $\widetilde{R}_i$ lie on the same side of 1, for $i = 1, 2$.

Let us adopt the notation

$$P_j(\mathbf{u}) \overset{\text{def}}{=} \Pr\{\mathbf{Y}_j = \mathbf{y}_j | \mathbf{U} = \mathbf{u}\}. \tag{5.2}$$

Then for $k = 2$, the transformations in (4.1) become

$$S_1^{(2m+1)} = \frac{P_1(10) + P_1(11)S_2^{(2m)}}{P_1(00) + P_1(01)S_2^{(2m)}} \tag{5.3}$$

$$S_2^{(2m+1)} = \frac{P_1(01) + P_1(11)S_1^{(2m)}}{P_1(00) + P_1(10)S_1^{(2m)}} \tag{5.4}$$

$$S_1^{(2m)} = \frac{P_2(10) + P_2(11)S_2^{(2m-1)}}{P_2(00) + P_2(01)S_2^{(2m-1)}} \tag{5.5}$$

$$S_2^{(2m)} = \frac{P_2(01) + P_2(11)S_1^{(2m-1)}}{P_2(00) + P_2(10)S_1^{(2m-1)}}. \tag{5.6}$$

To simplify the notation still further, we define

$$A_j \overset{\text{def}}{=} P_j(00), B_j \overset{\text{def}}{=} P_j(01), C_j \overset{\text{def}}{=} P_j(10), D_j \overset{\text{def}}{=} P_j(11). \tag{5.7}$$

If we combine (5.3) with (5.6), and (5.4) with (5.5), and use the notation in (5.7), we obtain, after some algebraic manipulations, for $m \geq 0$,

$$S_1^{(2m+1)} = \frac{(C_1A_2 + D_1B_2) + (C_1C_2 + D_1D_2)S_1^{(2m-1)}}{(A_1A_2 + B_1B_2) + (A_1C_2 + B_1D_2)S_1^{(2m-1)}} \tag{5.8}$$

$$S_2^{(2m+1)} = \frac{(B_1A_2 + D_1C_2) + (B_1B_2 + D_1D_2)S_2^{(2m-1)}}{(A_1A_2 + C_1C_2) + (A_1B_2 + C_1D_2)S_2^{(2m-1)}}. \tag{5.9}$$

Similarly we have for $m \geq 1$ that

$$S_1^{(2m)} = \frac{(C_2A_1 + D_2B_1) + (C_2C_1 + D_2D_1)S_1^{(2m-2)}}{(A_2A_1 + B_2B_1) + (A_2C_1 + B_2D_1)S_1^{(2m-2)}} \tag{5.10}$$

$$S_2^{(2m)} = \frac{(B_2A_1 + D_2C_1) + (B_2B_1 + D_2D_1)S_2^{(2m-2)}}{(A_2A_1 + C_2C_1) + (A_2B_1 + C_2D_1)S_2^{(2m-2)}}. \tag{5.11}$$

For definiteness, let us focus on (5.8). It says that $S^{(2m+1)}$ is obtained from $S^{(2m-1)}$ via the application of a linear fractional transformation with positive coefficients. It is easily shown that such a transformation has a unique positive fixed point. Furthermore, if we begin with any initial positive value, and repeatedly apply the transformation, the iterates will converge to the fixed point. This means that the sequence $S_1^1, S_1^3, S_1^5, \ldots$ will converge to the unique positive solution to the quadratic equation obtained from (5.8) when we set $S_1^{(2m+1)} = S_1^{(2m-1)} = x$, viz.,

$$(A_1C_2 + B_1D_2)x^2 + (A_1A_2 + B_1B_2 - C_1C_2 - D_1D_2)x - (C_1A_2 + D_1B_2) = 0. \tag{5.12}$$

From the form of (5.12), it follows that the roots are real and there is one positive and one negative root. Let Δ be the discriminant of (5.12), i.e., the positive square root of

$$\Delta^2 = (A_1A_2 + B_1B_2 - C_1C_2 - D_1D_2)^2 + 4(A_1C_2 + B_1D_2)(C_1A_2 + D_1B_2). \tag{5.13}$$

Then the positive root of (5.12) is

$$x_1^{\text{odd}} = \frac{(C_1C_2 + D_1D_2 - A_1A_2 - B_1B_2) + \Delta}{2(A_1C_2 + B_1D_2)}. \tag{5.14}$$

Thus

$$\lim_{m\to\infty} S_1^{(2m+1)} = x_1^{\text{odd}}.$$

Similarly, the sequence $S_1^2, S_1^4, S_2^6, \ldots$ converges to the unique positive root of the quadratic equation obtained by substituting x for both $S_1^{(2m)}$ and $S_1^{(2m-2)}$ in (5.10). This equation can be obtained (5.8) or (5.12) by interchanging the subscripts 1 and 2. But from the form of (5.12), it follows that when the subscripts 1 and 2 are interchanged, a root x is mapped into $-1/x$. Thus the unique positive fixed point of the linear fractional transformation (5.10) is the negative reciprocal of the negative root of (5.12), so that if we denote it by x_1^{even}, we have

$$x_1^{\text{even}} = \frac{2(A_1C_2 + B_1D_2)}{(A_1A_2 + B_1B_2 - C_1C_2 - D_1D_2) + \Delta}, \tag{5.15}$$

and so

$$\lim_{m\to\infty} S_1^{(2m)} = x_1^{\text{even}}.$$

It follows that the first component of the turbo decision vector $\widetilde{\mathbf{R}} = (\widetilde{R}_1, \widetilde{R}_2)$ defined in (4.2) is

$$\widetilde{R}_1 = x_1^{\text{odd}} x_1^{\text{even}} = \frac{(C_1C_2 + D_1D_2 - A_1A_2 - B_1B_2) + \Delta}{(A_1A_2 + B_1B_2 - C_1C_2 - D_1D_2) + \Delta}, \tag{5.16}$$

so that the turbo decision for u_1 is

$$\widehat{U}_1 = \begin{cases} 1 & \text{if } C_1C_2 + D_1D_2 \geq A_1A_2 + B_1B_2 \\ 0 & \text{if f } C_1C_2 + D_1D_2 < A_1A_2 + B_1B_2 \end{cases}$$

On the other hand, the OBD for U_1, according to (2.4), is based on the likelihood ratio

$$R_1 = \frac{P_1(10)P_2(10) + P_1(11)P_2(11)}{P_1(00)P_2(00) + P_2(01)P_2(01)} = \frac{C_1C_2 + D_1D_2}{A_1A_2 + B_1B_2}. \tag{5.17}$$

Thus the OBD for U_1, based on R_1, is exactly the same as the TBD for U_1, which is based on $\widetilde{R}_1$. An exactly similar argument shows that the OBD and TBD for U_2 also agree.

6. Algorithm Anomalies for $k = 3$.

In this section we will show by example that when $k \geq 3$, the turbo algorithm can behave unexpectedly. Here is a summary of the pathological behavior that can occur.

- *Example 1.* The turbo transformations have a unique fixed point, and the iterations converge to it, but the turbo decision does not agree with the OBD.
- *Example 2.* The transformations have multiple fixed points, one of which agrees with the OBD, but the iterations converge to a fixed point which does not agree with the OBD.
- *Example 3.* The limit in (4.2) does not exist; even the limit of the $\widehat{U}^{(m)}$ does not exist. Thus the turbo algorithm need not converge.

We will construct these examples (with $k = 3$) by carefully choosing the values $P_j(\mathbf{u})$ as defined in (5.2). In Appendix B, we will show that for at least one code, i.e., the $(7, 3)$ expurgated Hamming code, when the channel is the additive white Gaussian channel, for any 8 preassigned positive constants $L_{\mathbf{u}}$, for $\mathbf{u} = (000), \ldots, (111)$, there exists a potential received vector y such that

$$\Pr\{\mathbf{Y} = \mathbf{y} | \mathbf{U} = \mathbf{u}\} = \alpha L_{\mathbf{u}},$$

for some positive constant α. Since it is clear from the form of the turbo-transformations (3.1), that they are homogeneous in the $P_j(\mathbf{u})$'s, this validates our examples.

Example 1. In our first example we will take $k = 3$, and assume that the values $P_j(\mathbf{u})$ as defined in (5.2) do not depend on j. Then in simplified notation the $k = 3$ "turbo transformations" (cf. (3.1)) are

$$\begin{aligned} x_1' &= \frac{Ax_2x_3 + Bx_2 + Cx_3 + D}{Ex_2x_3 + Fx_2 + Gx_3 + H} \\ x_2' &= \frac{Ax_1x_3 + Bx_1 + Ex_3 + F}{Cx_1x_3 + Dx_1 + Gx_3 + H} \\ x_3' &= \frac{Ax_1x_2 + Cx_1 + Ex_2 + G}{Bx_1x_2 + Dx_1 + Fx_2 + H} \end{aligned} \tag{6.1}$$

We now impose the further condition that

$$G = A,\ H = B,\ E = C,\ F = D. \tag{6.2}$$

Then the equations in (6.1) become

$$\begin{aligned} x_1' &= \frac{(Ax_3 + B)x_2 + (Cx_3 + D)}{(Cx_3 + D)x_2 + (Ax_3 + B)} \\ x_2' &= \frac{(Ax_3 + B)x_1 + (Cx_3 + D)}{(Cx_3 + D)x_1 + (Ax_3 + B)} \\ x_3' &= \frac{(Ax_2 + C)x_1 + (Cx_2 + A)}{(Bx_2 + D)x_1 + (Dx_2 + B)} \end{aligned} \tag{6.3}$$

The symmetry in (6.3) suggests that x_1 and x_2 converge to 1 when the transformations are iterated. To prove that this is so, we use (6.3) to express $(x_1 - 1)/(x_1 + 1)$ and $(x_2 - 1)/(x_2 + 1)$ in terms of x_1, x_2, and x_3. The result is

$$\begin{aligned} \frac{x_1' - 1}{x_1' + 1} &= q(x_3)\frac{x_2 - 1}{x_2 + 1} \\ \frac{x_2' - 1}{x_2' + 1} &= q(x_3)\frac{x_1 - 1}{x_1 + 1}, \end{aligned} \tag{6.4}$$

where

$$q(x) = \frac{(A - C)x + (B - D)}{(A + C)x + (B + D)}.$$

Since $q(x)$ is monotonic, and since $x_3 \geq 0$, it follows that

$$|q(x_3)| \leq \lambda = \max\left(\frac{|A-C|}{|A+C|}, \frac{|B-D|}{|B+D|}\right).$$

Clearly $\lambda < 1$, so that by (6.4), $\lim(x_1 - 1)/(x_1 + 1) = \lim(x_2 - 1)/(x_2 + 1) = 0$, which implies $\lim x_1 = \lim x_2 = 1$, regardless of the initial values. What this means in terms of the turbo algorithm is that $\lim_{m\to\infty} S_1^{(2m-1)} = \lim_{m\to\infty} S_1^{(2m)} = \lim_{m\to\infty} S_2^{(2m-1)} = S_2^{(2m)} = 1$, so that the limit in (4.2) exists for the first and second components, and $\widetilde{R}_1 = \widetilde{R}_2 = 1$.

Looking now at the third equation in (6.3), we see that as x_1 and x_2 approach 1, x_3' converges to $(A+C)/(B+D)$. For the turbo algorithm, this means that $\lim_{m\to\infty} S_3^{(2m-1)} = \lim_{m\to\infty} S_3^{(2m)} = (A+C)/(B+D)$, so that the limit in the third component in (4.2) exists and is $((A+C)/(B+D))^2$. In summary, the limiting turbo likelihood estimate (cf. (4.2)) is

$$\widetilde{\mathbf{R}} = \left(1, 1, \left(\frac{A+C}{B+D}\right)^2\right). \tag{6.5}$$

On the other hand, from (2.4), the optimal likelihood ratios are

$$\mathbf{R} = \left(1, 1, \frac{A^2+C^2}{B^2+D^2}\right). \tag{6.6}$$

The third components in (6.5) and (6.6) are different, and it is easy to choose A, B, C, and D, so that the corresponding decisions are different. For example, if $A = 3$, $B = 5$, $C = 4$, $D = 1$, the turbo decision based on (6.5) is $(1,1,1)$ and the OBD based on (6.6) is $(1,1,0)$. This is the promised example of the turbo algorithm converging to a unique fixed point which gives the wrong decision. ▪

In Examples 2 and 3, we will again take $k = 3$ but now assume that the values $P_j(\mathbf{u})$ as defined in (5.2) depend on j, but only on the weight of $\mathbf{u}$. We shall use the notation

$$\begin{aligned} A_j &= P_j(111) \\ B_j &= P_j(011) = P_j(101) = P_j(110) \\ D_j &= P_j(001) = P_j(010) = P_j(100) \\ E_j &= P_j(000), \end{aligned} \tag{6.7}$$

for $j = 1, 2$.

Then in simplified notation, the $k = 3$ "turbo transformations" (cf. (3.1)) are

$$\begin{aligned} x_1' &= \frac{A_j x_2 x_3 + B_j(x_2 + x_3) + D_j}{B_j x_2 x_3 + D_j(x_2 + x_3) + E_j} \\ x_2' &= \frac{A_j x_1 x_3 + B_j(x_1 + x_3) + D_j}{B_j x_1 x_3 + D_j(x_1 + x_3) + E_j} \\ x_3' &= \frac{A_j x_1 x_2 + B_j(x_1 + x_2) + D_j}{B_j x_1 x_2 + D_j(x_1 + x_1) + E_j} \end{aligned} \tag{6.8}$$

The symmetry of (6.8) suggests that from any starting point, the iteration of the transformation (6.8) converges to the line defined by

$$x_1 = x_2 = x_3. \tag{6.9}$$

Let us prove that this is so, by using (6.8) to express $(x_1' - x_2')/(x_1 + x_2)$ in terms of x_1, x_2, and x_3. The result is a fraction in which all the terms in the denominator are positive. Dropping out some of these terms will increase the absolute value of the expression. In this way, we can obtain a bound of the form

$$\left|\frac{x_1' - x_2'}{x_1' + x_2'}\right| \le r_j(x_3) \left|\frac{x_1 - x_2}{x_1 + x_2}\right|, \tag{6.10}$$

where

$$r_j(x) = \left|\frac{(A_j x + B_j)(D_j x + E_j) - (B_j x + D_j)^2}{(A_j x + B_j)(D_j x + E_j) + (B_j x + D_j)^2}\right|. \tag{6.11}$$

It can be shown that there exist positive constants $\lambda_1 < 1$ and $\lambda_2 < 1$ such that for all $x \ge 0$, $|r_j(x)| \le \lambda_j$. Thus (6.10) becomes

$$\left|\frac{x_1' - x_2'}{x_1' + x_2'}\right| \le \lambda_j \left|\frac{x_1 - x_2}{x_1 + x_2}\right|. \tag{6.12}$$

This means that the quantity on the left of (6.12) decreases to zero exponentially under the iteration. A similar argument shows that the same is true for the corresponding expression in x_1 and x_3. It follows that whatever the starting values of (x_1, x_2, x_3), under iteration of the turbo transformation (6.8), the iterates will rapidly approach the line (6.9). Thus if there are any fixed points of the iterated turbo transformation, they must lie on the same line. In any case, we can assume that after a large number of iterations, we will have $x_1 = x_2 = x_3 = x$ and $x_1' = x_2' = x_3' = x'$ in (6.8). Then the three equations in (6.8) become the single equation

$$x' = \frac{A_j x^2 + 2B_j x + D_j}{B_j x^2 + 2D_j x + E_j}. \tag{6.13}$$

After this preliminary discussion, we are prepared to give Examples 2 and 3.

Example 2. In this example, we will assume that A_j, B_j, C_j, and D_j, as defined in (6.7) do not depend on j. In this case, it follows from the preceding discussion that the fixed points of the transformation (6.13) are the points (x, x, x) where x is a solution of the cubic equation

$$B_j x^3 + (2D_j - A_j)x^2 + (E_j - 2B_j)x - D_j = 0. \tag{6.14}$$

The equation (6.14) has at most three positive roots. If the roots are denoted r_1, r_2, and r_3, then from the expressions for the coefficients of a cubic as symmetric functions of the roots, we have

$$\begin{aligned} B &= 1 \\ D &= r_1 r_2 r_3 \\ A &= r_1 + r_2 + r_3 + 2r_1 r_2 r_3 \\ E &= r_1 r_2 + r_2 r_3 + r_3 r_1 + 2. \end{aligned} \tag{6.15}$$

We can assign any three positive values for r_1, r_2, and r_3, since by (6.15), the resulting values for A, B, D, and E will be positive. It is easy to show that if $r_1 < r_2 < r_3$, then if the initial value of x is less than r_2, the iterates will converge to r_1, whereas if the initial value of x is greater than r_2, the iterates will converge to r_3.

Now suppose that

$$r_1 < r_2 < r_3, \text{ and } r_1 r_3 = 1. \tag{6.16}$$

According to (2.4), and (6.7), in this case the optimal likelihood ratio is

$$R = R_1 = R_2 = R_3 = \frac{A^2 + 2B^2 + D^2}{B^2 + 2D^2 + E^2}. \tag{6.17}$$

Thus from (6.15) and (6.16), we have

$$R - 1 = \frac{A^2 + B^2 - D^2 - E^2}{B^2 + 2D^2 + E^2} = \frac{8 - (r_1 + r_3)^2}{B^2 + 2D^2 + E^2}(r_2^2 - 1). \tag{6.18}$$

Since $r_1 r_3 = 1$, the sum $r_1 + r_3$ can take any value greateer than 2. If we require $r_1 + r_3 < 2\sqrt{2}$, then $R - 1$, as expressed in (6.18), is greater than zero $\iff r_2 > 1$, and so $R > 1 \iff r_2 > 1$. However, by (4.1), the turbo algorithm starts at $x = 1$, and since $1 < r_2$, it follows from our previous discussion that the iterates will converge to r_2, leading to

$$\widetilde{\mathbf{R}} = (r_2^2, r_2^2, r_2^2) < (1, 1, 1).$$

Thus the OBD will be $(1, 1, 1)$ while the TBD will be $(0, 0, 0)$. ▪

Example 3. In this example, we will see that the limit in (4.2) need not exist. To get this behavior, we choose the coefficients in (6.13) so that the transformation is monotone decreasing for $j = 1$ and monotone increasing for $j = 2$. This implies that the composite function, which goes from one output of decoder 2 to the next, is a decreasing function. Such a function, mapping positive values into positive values, must have only one fixed point. If the derivative at this point has magnitude greater than 1, the fixed point is repulsive, since for small values of h,

$$|f(x + h) - f(x)| \approx |h||f'(x)| > |h|.$$

Thus convergence cannot occur.

A pair of transformations of the form (6.13) with these properties can be constructed in many ways. Here is one example:

$$\begin{aligned} A_1 &= .04, & B_1 &= 1, & D_1 &= 2, & E_1 &= .04 \\ A_2 &= 20, & B_2 &= 1, & D_2 &= 1, & E_2 &= 20. \end{aligned}$$

The composite function $f_1 \circ f_2$ has a fixed point at $x = 0.84246$, with derivative about -1.14. This will produce the desired effect. Indeed, using (4.1), and a computer, we find that $S^{(0)} = 1$, $S^{(1)} = .801587$, $S^{(2)} = .694696$, $S^{(3)} = 1.0325$, $S^{(4)} = 1.05427$, ..., and

$$\lim_{m \to \infty} S^{(4m+i)} = x_i, \quad \text{for } i = 0, 1, 2, 3, \tag{6.19}$$

where

$$x_0 = 8.34233, \quad x_1 = 0.208424, \quad x_2 = 0.111712, \quad x_3 = 4.45384. \tag{6.20}$$

Combining (6.19) with (4.4), we find that

$$\begin{aligned} \lim_{m \to \infty} \widetilde{R}^{(4m)} &= x_0 x_3 = 37.1554 \\ \lim_{m \to \infty} \widetilde{R}^{(4m+1)} &= x_1 x_0 = 1.73874 \\ \lim_{m \to \infty} \widetilde{R}^{(4m+2)} &= x_2 x_1 = .0232835 \\ \lim_{m \to \infty} \widetilde{R}^{(4m+3)} &= x_3 x_2 = .497547, \end{aligned} \tag{6.21}$$

which, by (4.5), implies that the limiting values of the turbo decision are

$$\widehat{U}^{(m)} = \begin{cases} 1 & \text{if } m \bmod 4 = 0 \text{ or } 1 \\ 0 & \text{if } m \bmod 4 = 2 \text{ or } 3. \end{cases}$$

Thus in this case the turbo decision algorithm does not converge. ▪

7. An Example Turbo-Code.

Consider the $(7,3)$ expurgated Hamming code with generator matrix

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}. \tag{7.1}$$

A simple "turbo code" based on this code is shown in Figure 3. The idea is that the three "information" bits U_1, U_2, and U_3 are encoded twice: once using the generator matrix G from (7.1), yielding the codeword $(X_1, X_2, \ldots, X_7)$, and once using πG, where π is a row permutation, yielding the codeword $(X_1', X_2', \ldots, X_7')$. Thus the encoder structure of Figure 3 represents, in effect, a $(14,3)$ linear block code.

We simulated the performance of this code on an additive white Gaussian channel (using a randomly selected permutation for each data point) using both an OBD algorithm, and a "turbo decision" algorithm, the idea being that the inputs to the turbo decision modules TDM 1 and TDM 2 of Figure 1 are the noisy versions of the two encoded versions of (U_1, U_2, U_3).

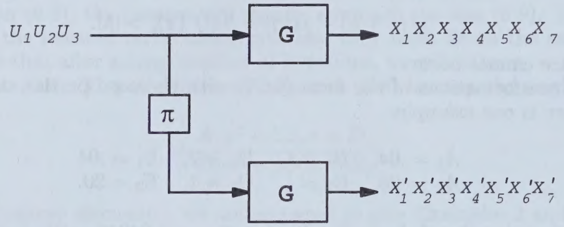


Figure 3. A $(14,3)$ turbo code. The matrix G is as in (7.1). The box labelled "π" is a random permutation of $\{1,2,3\}$.

The performance of this code on an additive white Gaussian channel, is shown in Figure 4. In Figure 4 are show three curves:

1. The performance of an OBD algorithm, which was computed using the formula (2.4).
2. The performance of one iteration of the TDA, i.e., one based on the likelihood vector $\widetilde{\mathbf{R}}^{(1)}$, as defined in (4.4).
3. The performance of 100 iterations of the TDA, i.e., one based on the likelihood vector $\widetilde{\mathbf{R}}^{(100)}$, as defined in (4.4).

We see from Figure 4 that despite the possibilities of anomalous behavior demonstrated in Section 6, the turbo algorithm nevertheless gives a remarkably good approximation to the OBD algorithm. The significance of this experiment is not that the $(14,3)$ turbo code is of any interest in itself, but rather that it is a turbo code which is small enough so that it is possible to implement the OBD, and directly compare it to the TDA. For "practical" turbo codes, for which an OBD algorithm is impossibly complex, this simulation, and other, similar, ones, suggests that the TDA approximates the OBD quite well, so that little potential performance is lost by using the sub-optimal TDA. We hope that future researchers will understand this phenomenon better and express themselves more articulately.

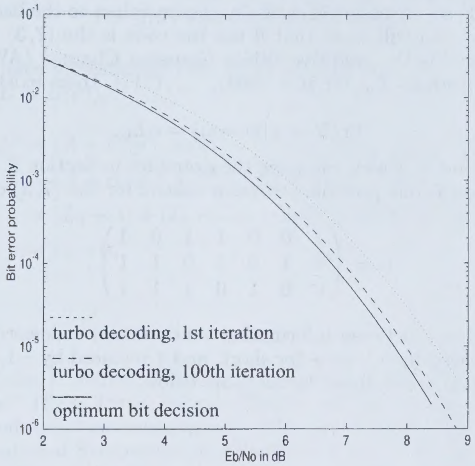


Figure 4. The performance of the $(14,3)$ turbo code described in Figure 3.

Appendix A. A Useful Lemma.

A.1 Lemma. *Suppose* $\mathbf{U} = (U_1, \ldots, U_k)$ *and* $\mathbf{Y} = (Y_1, \ldots, Y_n)$ *are random vectors defined on a common sample space* $\Omega = A^k \times B^n$. *Then for any* $a \in A$, $\mathbf{y} \in B^n$, *we have*

$$\Pr\{U_i = a | \mathbf{Y} = \mathbf{y}\} = \frac{\sum_{\mathbf{u}:\mathbf{u}_i = a} \Pr\{\mathbf{Y} = \mathbf{y} | \mathbf{U} = \mathbf{u}\} \Pr\{\mathbf{U} = \mathbf{u}\}}{\Pr\{\mathbf{Y} = \mathbf{y}\}}. \tag{A.1}$$

Proof: By Bayes' rule,

$$\Pr\{U_i = a | \mathbf{Y} = \mathbf{y}\} = \Pr\{U_i = a, \mathbf{Y} = \mathbf{y}\} / \Pr\{\mathbf{Y} = \mathbf{y}\}. \tag{A.2}$$

But by definition of a discrete sample space,

$$\Pr\{U_i = a, \mathbf{Y} = \mathbf{y}\} = \sum_{\mathbf{u}:\mathbf{u}_i = a} \Pr\{\mathbf{U} = \mathbf{u}, \mathbf{Y} = \mathbf{y}\}, \tag{A.3}$$

and by definition of conditional probability,

$$\Pr\{\mathbf{U} = \mathbf{u}, \mathbf{Y} = \mathbf{y}\} = \Pr\{\mathbf{Y} = \mathbf{y} | \mathbf{U} = \mathbf{u}\} \Pr\{\mathbf{U} = \mathbf{u}\}. \tag{A.4}$$

Combining (A.2), (A.3), and (A.4), we obtain (A.1). ▪

Appendix B. Likelihoods for the $(7,3)$ Expurgated Hamming Code.

In Section 6, we gave several examples of non-optimal behaviour of the turbo decoding algorithm for $k = 3$. In order to do so, we assigned carefully chosen values to the likelihoods $P_j(\mathbf{u})$ defined in (5.2). In this section, we will show that if the the code is the $(7,3)$ expurgated Hamming codes, and if the channel is the Additive White Gaussian Channel (AWGN), then for any 8 preassigned positive constants $L_{\mathbf{u}}$, for $\mathbf{u} = (000), \ldots, (111)$, there exists a potential received vector y such that

$$\Pr\{\mathbf{Y} = \mathbf{y} | \mathbf{U} = \mathbf{u}\} = \alpha L_{\mathbf{u}},$$

for some positive constant α, which validates the examples in Section 6.

As stated in Section 7, one possible generator matrix for the $(7,3)$ code is

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}. \tag{B.1}$$

Then if the correspondence between information words $\mathbf{u}$ and codewords $\mathbf{x}$ is $\mathbf{x} = \mathbf{u}G$, the 8 codewords (with 0 replaced by $+1$, or $+$ for short, and 1 replaced by -1, or $-$, as is customary for the Gaussian channel) are as listed in the table below.

$(B.2)$

i	u_i	x_i						
0	000	+	+	+	+	+	+	+
1	001	+	+	−	+	−	−	−
2	010	+	−	+	−	+	−	−
3	011	+	−	−	−	−	+	+
4	100	−	+	+	−	−	+	−
5	101	−	+	−	−	+	−	+
6	110	−	−	+	+	−	−	+
7	111	−	−	−	+	+	+	−

In general, if $\mathbf{y} = (y_1, \ldots, y_n)$ is received on the AWGN, the log likelihood that $\mathbf{x} = (x_1, \ldots, x_n)$ was transmitted is

$$\begin{aligned} \log p(\mathbf{y}|\mathbf{x}) &= \sum_{i=1}^{n} \log p(y_i | x_i) \\ &= -\sum_{i=1}^{n} \left(\frac{1}{2} \log(2\pi\sigma^2) + \frac{1}{2\sigma^2}(y_i - x_i)^2 \right) \qquad (B.3) \\ &= A(\sigma) + B(\sigma) \sum_{i=1}^{n} (y_i - x_i)^2, \end{aligned}$$

where $A(\sigma)$ and $B(\sigma)$ are constants which only depend on the noise variance σ and not on $\mathbf{x}$ or $\mathbf{y}$. The following Lemma is what we need.

B.1 Lemma. *If L_i, $i = 0, 1, \ldots, 7$ are any 8 preassigned constants, then for any value of σ, there is a vector $\mathbf{y}$ and a constant λ such that*

$$A(\sigma) + B(\sigma) ||\mathbf{y} - \mathbf{x}_i||^2 = L_i + \lambda, \tag{B.4}$$

for $i = 0, \ldots, 7$.

Proof: We first note that the columns in the array (B.2) are linearly independent, so that the rank of the array is 7. This means that there is exactly one linear dependence among the rows of the array, which is $\mathbf{x}_0 + \mathbf{x}_1 + \cdots + \mathbf{x}_7 = 0$. Thus the seven vectors $(\mathbf{x}_0 - \mathbf{x}_i)$, for $i = 1, \ldots, 7$, are linearly independent. Hence for any σ, there is a unique vector $\mathbf{y}$ such that

$$< \mathbf{y}, \mathbf{x}_0 - \mathbf{x}_i >= \frac{L_i - L_0}{2B(\sigma)}, \quad \text{for } i = 1, \ldots, 7. \tag{B.5}$$

With $\mathbf{y}$ chosen to satisfy (B.5), define λ by

$$(B.6) \qquad \lambda = A(\sigma) + B(\sigma)||\mathbf{y} - \mathbf{x}_0||^2 - L_0.$$

Then from (B.6) we have immediately that (B.4) is true for $i = 0$. For $i = 1, \ldots, 7$, we have (with $A = A(\sigma)$ and $B = B(\sigma)$),

$$\begin{aligned} A + B||\mathbf{y} - \mathbf{x}_i||^2 &= (A + B||\mathbf{y} - \mathbf{x}_0||^2) + (A + B||\mathbf{y} - \mathbf{x}_i||^2) - (A + B||\mathbf{y} - \mathbf{x}_0||^2) \\ &= (A + B||\mathbf{y} - \mathbf{x}_0||^2) + 2B < \mathbf{y}, \mathbf{x}_0 - \mathbf{x}_i > \\ &= (L_0 + \lambda) + (L_i - L_0) \quad \text{(from (B.6) and (B.5))} \\ &= L_i + \lambda, \end{aligned}$$

as asserted. ▪

References.

1. L. R. Bahl, J. Cocke, F. Jelinek, J Raviv, "Optimal decoding of linear codes for minimizing symbol error rate." *IEEE Trans. Inform. Theory*, vol. IT-20 (March 1974), pp. 284–287.
2. G. Battail, "Pseudo-random recursive convolutional coding for near-capacity performance," Proc. 2nd International Symposium on Communication theory and Applications, Ambleside, U.K., July 1993, pp. 54–65.
3. G. Battail, "Can we implement random coding?" in *Codes and Cyphers: Cryptography and Coding IV (Proc. 4th IMA Conference on Cryptography and Coding, Cirencester, U.K., December 1993)*, P. G. Farrell, ed., pp. 1–15.
4. S. Benedetto and G. Montorsi, "Average performance of parallel concatenated block codes." *Electronics Letters*, vol. 31, no. 3 (2 Feb. 1995), pp. 156–158.
5. S. Benedetto and G. Montorsi, "Performance evaluation of turbo codes." *Electronics Letters*, vol. 31, no. 3 (2 Feb. 1995), pp. 163–165.
6. S. Benedetto and G. Montorsi, "Role of recursive convolutional codes in turbo codes." *Electronics Letters*, vol. 31, no. 11 (25 May 1995), pp. .
7. G. Berrou, A. Glavieux, P. Thitimajshima, "Near Shannon limit error-correcting coding: Turbo codes," *Proc. 1993 International Conf. Comm.*, (Geneva, May 1993), pp. 1064–1070.
8. J.-F. Cheng and R. J. McEliece, "Unit memory Hamming turbo codes," *Proc. 1995 ISIT*, *in press.*
9. D. Divsalar and F. Pollara, "Turbo codes for deep-space communications," *TDA Progress Report* vol. 42-120 (February 15, 1995), pp. 29–39.
10. D. Divsalar and F. Pollara, "Multiple turbo codes for deep-space communications," *TDA Progress Report* vol. 42-121 (May 15, 1995), *in press.*
11. D. Divsalar and F. Pollara, "Turbo codes for PCS applications," *Proc. ICC '95* (June 18–22, 1995), *in press.*
12. D. Divsalar, S. Dolinar, R. J. McEliece, and F. Pollara, "Transfer function bounds on the performance of turbo codes," *TDA Progress Report* vol. 42-122 (July 15, 1995), *in press.*
13. D. Divsalar, S. Dolinar, R. J. McEliece, and F. Pollara, "Performance analysis of turbo codes," submitted to *MILCOM 1995.*
14. S. Dolinar and D. Divsalar, "Weight distributions for turbo codes using random and non-random interleaving." *TDA Progress Report* vol. 42-122 (July 15, 1995), *in press.*
15. J. Hagenauer and P. Robertson, "Iterative (Turbo) decoding of systematic convolutional codes with the MAP and SOVA algorithms," Proc. ITG conf. "Source and Channel Coding," Oct. 1994, Frankfurt, Germany.
16. J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," to appear, *IEEE Trans. Inform. Theory.*
17. P. Robertson, "Illuminating the structure of code and decoder of parallel and concatenated recursive systematic (Turbo) codes," Proc. GLOBECOM 1994, pp. 1298–1303.

Convolutional Codes over Groups: A Pragmatic Approach

Thomas Mittelholzer
Signal and Information Processing Laboratory
Swiss Federal Institute of Technology
CH-8092 Zurich, Switzerland

Abstract *A convolutional code over a group (CCG) is defined as a time-invariant group code that is strongly observable and strongly controllable. This definition is well-suited to different choices of the code sequences, such as finite-weight, one-sided infinite and two-sided infinite code sequences. It is shown that a CCG is uniquely characterized by its canonical trellis. For a convolutional code over a locally compact abelian group, the dual code – which is again a CCG – is defined via the dual of its canonical trellis.*

1. Convolutional Codes

Let G be a group and let G^{Z} denote the direct product group whose components are indexed by the integers Z. A subgroup $C \subseteq G^{\mathrm{Z}}$ is called a *group code* over G. The time-j component of a sequence $\mathbf{c} \in G^{\mathrm{Z}}$ will be denoted as $\mathbf{c}(j)$ and σ will denote the shift operator on G^{Z}, defined by $\sigma(\mathbf{c})(j) = \mathbf{c}(j+1)$. C is *time-invariant* if $\sigma(C) = C$. We recall from [1] the notions of observability and controllability of codes, which are important for the definition of a convolutional code; for other notions such as (ℓ-)completeness, canonical state spaces, canonical trellises, observability and controllability of trellises, we refer the reader to [1].

A code C is called *ℓ-controllable* if for any time instant $j \in \mathrm{Z}$ any two code sequences $\mathbf{c}$ and $\mathbf{c}'$ can be 'joined' by a code sequence $\mathbf{c}''$ within a time span ℓ, that is, $\mathbf{c}$ and $\mathbf{c}''$ coincide for all times $i < j$ and $\mathbf{c}'$ and $\mathbf{c}''$ coincide for all times $j + \ell \leq i$. The code C is called *strongly controllable* if it is ℓ-controllable for some nonnegative integer ℓ. A code C is called *ℓ-observable* if for any time instant $j \in \mathrm{Z}$ and any two code sequences $\mathbf{c}$ and $\mathbf{c}'$ that coincide on the time interval $\{j, j+1, \ldots, j+\ell-1\}$, the concatenation of the left-sided sequence $\mathbf{c}(i)$, $i<j$ with the right-sided sequence $\mathbf{c}'(i)$, $j \leq i$, is also a code sequence. The code C is called *strongly observable* if it is ℓ-observable for some nonnegative integer ℓ.

Definition A *convolutional code over* G (CCG) is a time-invariant group code in G^{Z} that is strongly controllable and strongly observable.

Examples 1.) The set of all binary sequences of finite and even weight form a time-invariant group code C_{even} but not a convolutional code over the binary group, because C_{even} is not strongly observable. The statements (i) and (iii) of Theorem 1 (below) do not apply.

2.) Every standard (n, k)-convolutional code over some field $\mathbb{F}$ can be viewed as a convolutional code over the additive group of the vector space $\mathbb{F}^n$. Note that it is a CCG regardless of whether the code sequences are taken to be finite-weight, rational, Laurent or two-sided infinite sequences.

2. Local Description of Convolutional Codes

The *canonical state space* of a time-invariant group code C is defined as the quotient group

$$S_C = C/(C_- + C_+)$$

where $C_+ = \{\mathbf{c} \in C : \mathbf{c}(i) = 0 \text{ for } i < 0\}$ and $C_- = \{\mathbf{c} \in C : \mathbf{c}(i) = 0 \text{ for } i \geq 0\}$ [here additive notation is used although the groups are not assumed to be abelian]. Every code sequence $\mathbf{c}$ determines an equivalence class $s = [\mathbf{c}]$ in the quotient group S_C. The *canonical branch group* B_C of C, which is a subgroup of $S \times G \times S$, is defined by

$$B_C = \{([\mathbf{c}], \mathbf{c}(0), [\sigma(\mathbf{c})]) : \mathbf{c} \in C\}.$$

The canonical branch group determines a time-invariant trellis by taking its elements (s, g, t) as branches that connect a current state s with the next state t in the trellis and have branch label g (cf. [1]). The resulting trellis $\mathcal{X}_C$ is called the *canonical trellis* of C. Let $\Lambda(\mathcal{X}_C)$ denote the set of all label sequences in $G^{\mathbb{Z}}$ that are obtained from bi-infinite paths through the trellis $\mathcal{X}_C$. The set $\Lambda_f(\mathcal{X}_C)$ is defined likewise with the additional restriction that all paths have finite support, i.e., for each path all but a finite number of its branches are zero. $\Lambda(\mathcal{X}_C)$ and $\Lambda_f(\mathcal{X}_C)$ are time-invariant group codes generated by B_C. The following theorem shows that the local description B_C of a CCG remains invariant under different choices of sequence spaces such as finite-weight and two-sided infinite sequence spaces.

Theorem 1 Let C be a convolutional code over a group G. Then,
(i) the canonical trellis $\mathcal{X}_C$ is strongly controllable and strongly observable,
(ii) the code $\Lambda(\mathcal{X}_C)$ is equal to the topological completion of C in $G^{\mathbb{Z}}$,
(iii) the canonical branch groups of $\Lambda_f(\mathcal{X}_C)$ and $\Lambda(\mathcal{X}_C)$ are both isomorphic to B_C.

Recently, a duality theory has been developed for group codes [2] that makes use of Pontryagin duality (cf. [3]). The simple local description of a CCG by its canonical trellis leads us to define duality locally on the canonical branch group. This can be viewed as a pragmatic approach that avoids finiteness issues that are involved in the definition of a dual of a two-sided infinite CCG (cf. [2]). For a locally compact abelian group G, let G' denote the continuous character group, i.e., G' consists of all continuous homomorphisms g' from G into the 1-dimensional torus $\mathbb{R}/\mathbb{Z}$ (cf. [3]).

Definition. The *dual branch group* $B_C^{\perp}$ is defined by

$$B_C^{\perp} = \{(s', g', t') \in S' \times G' \times S' : s'(s) + g'(g) - t'(t) = 0 \text{ in } \mathbb{R}/\mathbb{Z} \text{ for all } (s, g, t) \in B_C\}.$$

Theorem 2 Let B_C be the canonical branch group of a convolutional code C defined over a locally compact abelian group G. If B_C is closed in $S \times G \times S$, then the trellis generated by $B_C^{\perp}$ generates a CCG over the character group G'. Moreover, $(B_C^{\perp})^{\perp} = B_C$.

References

[1] H.-A. Loeliger, G. D. Forney, Jr., T. Mittelholzer, M. D. Trott, 'Minimality and observability of group systems', *Lin. Algebra & Appl.*, vol. 205–206, pp. 937–963, July 1994.

[2] G. D. Forney and M. D. Trott, 'The dynamics of group codes: dual group codes and systems,' Manuscript Sep. 1994.

[3] E. Hewitt and K.A. Ross, *Abstract harmonic analysis*, Vol. I, Springer 1963.

Ascetic convolutional codes

Øyvind Ytrehus

University of Bergen, Department of Informatics,
Høyteknologisenteret, N-5020 Bergen, Norway
E-mail: oyvind@ii.uib.no

Supported by the Norwegian Research Council (NFR).

Acknowledgement

The "Applications" section of this paper presents an overview of research carried out at the Department of Informatics at the University of Bergen. Kjell Jørgen Hole, Marianne Fjelltveit Hole, and Rolf Erstad made significant contributions to this work.

Abstract

Ascetic convolutional codes contain fewer edge labels than other convolutional codes. This is advantageous in several practical situations. Selecting an ascetic code over a non-ascetic code carries no severe penalty, in terms of trade-off of minimum distance, code rate and complexity.

1. Introduction

One day at the dawn of time, when convolutional codes were created, all the little convolutional code children were assigned the following task: They were to assign binary labels to their so far naked trellis edges, in order to achieve a Large Free Hamming Distance.

Every little $(n = 8, k = 6, \nu = 1)$ convolutional code distributed vectors from all of F_2^n over their trellis edges. Ah, well, not quite everyone. One little code only selected even-weight labels. All the other little codes chided her, "How can you *possibly* expect to achieve a Large Free Hamming Distance when you don't use more than half of the available edge labels? Here, try (01111111)". But she answered, stubbornly, "No. I need no odd-weight vectors".

She was an *ascetic convolutional code.*

Judgement day arrived, and the little codes had to submit their (now edge labelled) trellis to the Modified Viterbi Algorithm. Some clever codes had achieved Free Hamming Distances of one or two or even three, some careless ones had constructed catastrophic trellises and were forever doomed to Loss of Dimension or even to an existence as mere block codes. "Was this the last one?", asked the Modified Viterbi Algorithm, preparing to leave, (it had been a long day). The little codes answered, "Well, there's that little ascetic code, but she only used half of the edge labels anyway". As usual in this kind of story, the little ascetic code still had a go with the Modified Viterbi Algorithm. What was her Free Hamming Distance? The answer is postponed until Theorem 3.5.

(But she did live happily ever after).

2. Preliminaries

The notation introduced in this section is kept at a minimum. In particular, we shall neither need nor define the concepts of *encoder*, *basic encoder*, *minimal encoder*, or the procedure for obtaining a minimal encoder from the parity check matrix. For a comprehensive treatment of the theory of convolutional codes the reader is referred to [1], or to the pioneering work in [2, 3, 4].

Let $\mathbf{h} = (h_1, \dots, h_t)^T$ (for some $t > 0$) be a binary (column) vector. The *l-th shift* of $\mathbf{h}$ is the sequence $\mathbf{h}^{\overset{l}{\rightarrow}} = \left(\overbrace{0, \dots, 0}^{l}, h_1, h_2, \dots, h_t, 0, \dots\right)^T$, starting at position one and extending to infinity, in which $\mathbf{h}$ can be found as the subvector starting in the $(l+1)$-th position and all other entries are zero.

Let similarly $\mathbf{H}$ be a binary $(\nu + 1) \times n$ matrix. The l-th shift of $\mathbf{H}$ is the matrix $\mathbf{H}^{\overset{l}{\rightarrow}}$ of n columns and an infinite number of rows, in which $\mathbf{H}$ can be found as the submatrix starting in the $(l+1)$-th row and all other entries are zero. Let further

$$\begin{aligned}\mathbf{H}^* &= (\mathbf{H}^{\overset{0}{\rightarrow}} \quad \mathbf{H}^{\overset{1}{\rightarrow}} \quad \mathbf{H}^{\overset{2}{\rightarrow}} \quad \cdots) \\ &= (\mathbf{h}_1^* \quad \mathbf{h}_2^* \quad \cdots \quad \mathbf{h}_n^* \quad \mathbf{h}_{n+1}^* \quad \cdots),\end{aligned}$$

where for $i \geq 1$, the column $\mathbf{h}_i^* = \mathbf{h}_{((i-1) \bmod n)+1}^{\overset{\lfloor (i-1)/n \rfloor}{\rightarrow}}$ is the $\lfloor (i-1)/n \rfloor$-th shift of the $(((i-1) \bmod n) + 1)$-th column of $\mathbf{H}$.

Let $\mathcal{F}$ be the set of infinite binary sequences $\mathbf{u} = (u_1, u_2, \dots)$ starting at time 1. $\mathbf{H}$ is a parity-check matrix for a convolutional code $\mathcal{C} = \{\mathbf{u} \in \mathcal{F} | \mathbf{H}^* \mathbf{u}^T = \mathbf{h}_1 u_1 + \mathbf{h}_2 u_2 + \dots + \mathbf{h}_i u_i + \dots = \mathbf{0}\}$. The elements of $\mathcal{C}$ are called code words.

(***Remark.*** Strictly speaking, $\mathcal{C}$ should be defined as a vector space over the field of Laurent series over $F = GF(2)$ (see e. g. [1]). The current definition was chosen to enhance simplicity and since in practice nonzero code words will start at or after some fixed time.)

$\mathcal{C}$ has *block length* n, *dimension* $n - 1$ and constraint length at most ν. The constraint length is equal to ν if there exists no other parity check matrix $\mathbf{H}'$ for the same code $\mathcal{C}$ with smaller row dimension; in this case $\mathbf{H}$ is a minimal parity check matrix. In the following we will assume that parity check matrices are minimal.

Let $\mathcal{H}$, referred subsequently to as a *combined* matrix, be a collection of r matrices $\mathbf{H}^{(i)}, i = 1, \dots, r$, over F where $\mathbf{H}^{(i)}$ has $\nu_i + 1$ rows and n columns.

The n columns of $\mathbf{H}^{(i)}$ are denoted $\mathbf{h}_j^{(i)} = (h_{j,0}^{(i)}, h_{j,1}^{(i)}, \dots, h_{j,\nu_i}^{(i)})^T$, $j = 1, \dots, n$, where the last row of $\mathbf{H}^{(i)}$ has at least one nonzero entry. The binary column vector $\mathbf{h}_j = (h_{j,0}^{(1)}, \dots, h_{j,\nu_1}^{(1)}, h_{j,0}^{(2)}, \dots, h_{j,0}^{(r)}, \dots, h_{j,\nu_r}^{(r)})^T \in F^{\nu + r}$ will be called the j-th *combined column* of $\mathcal{H}$. Thus $\mathcal{H}$ can be regarded as a binary $\left(\sum_{i=1}^{r} (\nu_i + 1)\right) \times n$ matrix.

If there are too many indices floating around here, it might be time for an example.

Example 1. Let

$$\mathbf{H}' = \left(\begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ \hline 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{array}\right).$$

The code $\mathcal{C}_1$ defined by this matrix has length 8 and dimension 4.

□

(A more common notation would be to represent $\mathbf{H}$ in terms of its *D-transform* $\mathbf{H}(D)$).

Example 1 (Cont.). The D-transform of $\mathbf{H}'$ is

$$\mathbf{H}'(D) = \left(\begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1+D & D & 1 & D & D & 1+D & 1 & 0 \\ \hline D & 1 & 1 & D & 0 & 1+D & 0 & 1+D \\ \hline 1 & D & 1 & 1 & 0 & D & D & 1+D \end{array}\right).$$

□

Each matrix $\mathbf{H}^{(r)}$ is a parity-check matrix for some convolutional code $\mathcal{C}^{(i)}$ of dimension $(n-1)$. We call $\mathcal{H}$ the parity-check matrix for the convolutional code $\mathcal{C} = \cap_{i=1}^{r}\mathcal{C}^{(i)}$. $\mathcal{C}$ has block length n, dimension at least $n-r$, and *constraint length* (or *memory*) $\nu = \sum_{i=1}^{r}\nu_i$. The *free distance* $d_{\text{free}} = d_{\text{free}}(\mathcal{C})$ is the minimum Hamming weight of any nonzero code word in $\mathcal{C}$. $\mathcal{C}$ is said to be an $(n, n-r, \nu, d_{\text{free}})$ binary convolutional code. If we want to emphasize the distribution of the ν_is, the code will be referred to as an $(n, n-r, \underline{\nu}, d_{\text{free}})$ code, where $\underline{\nu} = (\nu_1, \ldots, \nu_r)$ is the constraint length vector.

Example 1 (Cont.) The code $\mathcal{C}_1$ has constraint length vector (0,1,1,1) and constraint length 3. The free Hamming distance of $\mathcal{C}_1$ is, incidentally, 8. Thus, it is possible to find 8 columns of the associated doubly semi-infinite matrix $\mathbf{H}^*$ which are linearly dependent, but any 7 or less columns of $\mathbf{H}^*$ are linearly independent.

□

Assumption. Without loss of generality, we can assume that $\nu_1 \leq \ldots \leq \nu_r$. For a particular code, the parity check matrix is not unique, but the sequence $\nu_1, \ldots, \nu_r$ is [1].

Definition. $N(r, (\nu_1, \ldots, \nu_r), d_{\text{free}})$ is the largest n such that there exists a convolutional code of block length n, dimension $n-r$, constraint length vector $(\nu_1, \ldots, \nu_r)$, and free distance d_{free}. Similarly, $N(r, \nu, d_{\text{free}}) = \max_{(\nu_1+\ldots+\nu_r=\nu)} N(r, (\nu_1, \ldots, \nu_r), d_{\text{free}})$. We assume that $d_{\text{free}} \geq 3$; then $N(r, \nu, d_{\text{free}})$ is finite.

Definition. An $(n, n-r, \underline{\nu}, d_{\text{free}})$ convolutional code $\mathcal{C}$ is *ascetic* if not all binary vectors in F_2^n can be found as edge labels in the trellis for $\mathcal{C}$. It is easy to see, and shown in [5], that the set of edge labels is a vector space of dimension $n - |\{i, 1 \leq i \leq r : \nu_i = 0\}|$. Thus, in particular, ascetic codes have $\nu_1 = 0$.

3. The Free Hamming distance of ascetic codes

In this section we discuss Hamming distance properties of ascetic codes, and compare these codes to non-ascetic codes.

The best known upper bounds on $N(r, \nu, d_{\text{free}})$ were established by considering the block codes obtained by truncation of the convolutional code. The following theorem is a slight generalization ([6]) of a result appearing in [7].

Theorem 3.1 ([7]). *Let* $\mathcal{C}$ *be an* $(n, n-r, \nu, d_{\text{free}})$ *binary convolutional code defined by the combined parity check matrix* $\mathcal{H}$ *with combined columns* $\mathbf{h}_j, j = 1, \ldots, n$. *Let* $\varepsilon_j, 1 \leq j \leq n$, *be defined by* $\varepsilon_j = max\{s | h^{(i)}_{j,\nu_i - l} = 0 \text{ for } 1 \leq i \leq r, 0 \leq l \leq s\}$. *Then*

$$d_{\text{free}} \leq \min \left\{ d(N, K) \mid N = mn + \sum_{j=1}^{n} \varepsilon_j,\ 1 \leq K = N - (\nu + rm),\ m \geq 0 \right\}, \quad (1)$$

where $d(N, K)$ *is the largest minimum Hamming distance of any linear binary block code of block length* N *and dimension* K. *(Note: A table of bounds on* $d(N, K)$ *can be found in [8]).*

3.1. Bounds and constructions for small free Hamming distances

It turns out (cf. Theorem 3.5) that the choice of $r = 1$ (which if $\nu > 0$ implies that the code is non-ascetic) is a poor one when we are shooting for small free Hamming distances:

Lemma 3.2 ([6]) $N(1, \nu, 4) = 2^{\nu - 1}$.

Similarly, we have the following lemma which applies to codes with $r > 1$:

Lemma 3.3. *For* $r \geq 2$, $N(r, \nu, 4) \leq 2^{\nu + r - 1}$ *([6]).*

Simple Proof. If there existed a $(2^{\nu+r-1} + 1, 2^{\nu+r-1} + 1 - r, \nu, 4)$ code, it could be truncated (as in the Heller bound) to a $[2^{\nu+r-1} + 1, 2^{\nu+r-1} + 1 - (\nu + r), 4]$ block code, which in turn could be punctured to a $[2^{\nu+r-1}, 2^{\nu+r-1} + 1 - (\nu + r), 3]$ block code. This contradicts the Hamming bound.

□

We observe that if a combined matrix contains a combined column $\mathbf{h}$ as well as some shift $\mathbf{h}^{\overset{l}{\rightarrow}}$ of $\mathbf{h}$, then the corresponding convolutional code will have minimum distance at most two. We will therefore avoid the use of combined columns $\mathbf{h} = (h_0^{(1)}, \ldots, h_{\nu_1}^{(1)}, \ldots, h_0^{(r)}, \ldots, h_{\nu_r}^{(r)})^T$ such that

$$h_0^{(i)} = 0, \ \forall i \in \{1, \ldots, r\}. \quad (2)$$

Theorem 3.4. *For* $\nu \geq 1$ *and* $r \geq 1$, $N(r, \nu, 3) = (2^r - 1)2^\nu$.

Proof. Wyner and Ash [9] noted that an $(n, n-1, \nu, 3)$ code can be constructed by selecting the parity check matrix $\mathbf{H}$ as the matrix consisting of all distinct $(\nu + 1)$-dimensional vectors $\mathbf{h} = (h_0, \ldots, h_\nu)^T$ that have first entry $h_0 = 1$.
For $r > 1$, we can similarly choose $\mathcal{H}$ as the matrix in which the set of combined columns consists of all distinct $(\nu + r)$-dimensional vectors $\mathbf{h} = (h_0^{(1)}, \ldots, h_{\nu_1}^{(1)}, \ldots, h_0^{(r)}, \ldots, h_{\nu_r}^{(r)})$ except those on the form (2). The number of such vectors is $2^{\nu+r} - 2^\nu$.

□

Note that there is no restriction in this construction on the distribution of the ν_is. In particular, optimal codes of distance 3 can be constructed as "very ascetic", i. e., $\nu_1 = \cdots = \nu_{r-1} = 0, \nu_r = \nu$.

Theorem 3.5. *For* $\nu \geq 1$ *and* $r > 1$, $N(r, \nu, 4) = 2^{\nu+r-1}$.

Proof. "$\leq$" was shown in Lemma 3.3.
"$\geq$:" A $(n, n-r, \nu, 4)$ code can be constructed as follows. Let the first element ν_1 of the constraint length vector be zero, the remainder of the constraint length vector can be chosen arbitrarily subject to $\sum_{j=2}^{r} \nu_j = \nu$. Select the combined parity check matrix $\mathcal{H}$ such that it consists of all distinct combined columns $\mathbf{h}$ that have "1" as their first entry. Since the columns are distinct the free distance is at least 3, and since the first row of $\mathcal{H}$ is the all-one vector and $\nu_1 = 0$, each code word has even Hamming weight in each block.

□

We note that the codes described in Theorems 3.4 and 3.5 can be thought of as the convolutional code counterpart to the Hamming codes and its even weight subcodes.

Theorem 3.6 *For* $\nu \geq 1$ *and* $r > 1$, *non-ascetic codes of free Hamming distance 4 have length* $n < 2^{\nu+r-1}$.

In order to prove this, we shall resort to a different proof of Lemma 3.3.

Alternative Proof of Lemma 3.3 ([6]). Let $\mathcal{H}$ define an $(n, n-r, \nu, 4)$ binary convolutional code. Let $X_i, i = 1, \ldots, 2^r - 1$ be the set of combined columns $\mathbf{h}_j$ in $\mathcal{H}$ such that $(h_{j,0}^{(1)}, \ldots, h_0^{(r)})$ is the binary expansion of i, and let $x_i = |X_i|$. W. l. o. g., we can assume that $2^\nu \geq x_1 \geq x_i, 2 \leq i \leq 2^r - 1$. We first show that

$$\forall i : 1 \leq i \leq 2^{r-1} - 1 : x_{2i} + x_{2i+1} \leq 2^\nu. \tag{3}$$

If $x_{2i} = 0$, there is nothing to prove, so assume that $x_{2i} > 0$. Since the free distance is more than three, it follows that if $\mathbf{a} \in X_1$ and $\mathbf{b} \in X_{2i}$, then $(\mathbf{a} + \mathbf{b}) \notin X_{2i+1}$. There are x_1 choices for $\mathbf{a}$, hence $x_{2i+1} \leq 2^\nu - x_1$ implying (3). So

$$n = \sum_{j=1}^{2^r - 1} x_j = x_1 + \sum_{i=1}^{2^{r-1}-1} (x_{2i} + x_{2i+1}) \leq x_1 + (2^{r-1} - 1)2^\nu \leq 2^{\nu+r-1}.$$

□

Proof of Theorem 3.6 From the alternative proof of Lemma 3.3, it follows that if the block length of the code is $n < 2^{\nu+r-1}$, then $x_1 = 2^\nu$. Hence, the combined parity check matrix contains combined columns $\mathbf{x} = (1, 0, 0, 0, \ldots, 0)^T$ and $\mathbf{y} = (1, 1, 0, 0, \ldots, 0)^T$. If $\nu_1 > 0$, then $\mathbf{x} + \mathbf{y} + \mathbf{x}_{0}^{\overset{1}{\rightarrow}} = \mathbf{0}$, thus the minimum distance is at most 3.

□

3.2. Computer search results

In the previous section, we gave explicit constructions of codes with small free Hamming distances. It is harder to *construct* convolutional codes of free Hamming distance larger than 4. This applies to ascetic as well as non-ascetic codes.

A computer search (described in [6]) has been used to find lower bounds (non-exhaustive "tabu" search) and exact values (exhaustive search) on $N(r, \nu, d_{\text{free}})$ and $N(r, \nu, d_{\text{free}})$ for

$r \leq 4$ and moderate values of ν and d_{free}. Some results were presented in [10, 11]. As a general empirical observation, $N_A(r, \nu, d_{\text{free}}) = \max_{(\nu_1 = 0, \nu_2 + \ldots + \nu_r = \nu)} N(r, (\nu_1, \ldots, \nu_r), d_{\text{free}})$ is usually close to or equal to $N(r, \nu, d_{\text{free}})$.

Sometimes surprisingly good "very ascetic" codes can be found.

Example 2. *The code C_1 in Example 1 is ascetic and known to be optimal, in the sense that its block length is equal to $N(4, 3, 8)$. However, a more ascetic code exists. Let*

$$\mathbf{H}'' = \left(\begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ \hline 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ \hline 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \end{array}\right)$$

define the code C_2. C_2 is an (8,4,(0,0,0,3),8) code.

To be fair, there is a penalty for the escetic behaviour of this code: while C_1 has 29 codewords of weight 8 and *truncation length* 5, while C_1 has 45 codewords of weight 8 and truncation length 10.

4. Applications

This section describes some applications of ascetic codes.

4.1. Zero-run lengths [12, 11]

In many communication and recording systems, the receiver derives symbol synchronization information from the transmitted or recorded sequence. If the symbol "zero" is represented as "absence of signal", a long zero-run (sequence of consecutive zeros) may cause the receiver to lose synchronization, and should be avoided. Hence, the maximum zero-run length, L_{max}, of a code is a design parameter which should be as small as possible.

Using a result by D. Forney, Hole [12] showed that for any coset of any convolutional code, $L_{max} \geq \nu_1$. Also, for any convolutional code there is at least one coset for which $L_{max} \leq (\nu_1 + 2)n - 2 - \epsilon$, where ϵ is a nonnegative integer depending on the code. Thus ascetic codes are preferable for these purposes, since $\nu_1 = 0$ and, thus, $L_{max} \leq 2n - 2 - \epsilon$. Note that this upper bound is independent on the constraint length ν. Further results on this problem was recently presented by Hole and Ytrehus [11].

4.2. Codes for precoded partial-response channels [5, 13, 14]

Wolf and Ungerboeck [15] suggested the use of cosets of convolutional code cosets for coding on precoded $1 - D$ partial response channels. In the absence of noise, on a precoded $1 - D$ channel, a binary zero input results in a zero output, while a "1" input results in a "+1" or a "-1" output, where the signs alternate. When Gaussian noise is added at the channel output, the relevant distance measure is the squared Euclidean distance, defined as $d^2(\mathbf{u},\mathbf{v}) = \sum_i (u_i - v_i)^2$, where $\mathbf{u} = (u_0, u_1, \ldots)$ and $\mathbf{v} = (v_0, v_1, \ldots)$ are output sequences. Wolf and Ungerboeck noted that the free squared Euclidean distance of a code is lower bounded by its Hamming distance. They gave examples of codes, using $(n, n-1)$ convolutional codes with large free Hamming distance. However they were unable to provide examples with Euclidean distance larger than the lower bound.

The reason for this is that cosets of non-ascetic codes contain long zero-runs. It is easy to see that the squared Euclidean distance between output sequences $\mathbf{u} = (u_0, u_1, \ldots)$ and $\mathbf{0} = (0, 0, \ldots)$ is equal to $d^2(\mathbf{u}, \mathbf{0}) = \sum_i (u_i)^2 = \sum_i |u_i|$ = the Hamming distance between $\mathbf{u}$ and $\mathbf{0}$. On the other hand, many ascetic code cosets provide a free squared Euclidean distance which is much larger than the free Hamming distance of their corresponding codes.

Example 3. *Consider the code C_3 defined by the parity check matrix*

$$\mathbf{H}''' = \begin{pmatrix} 0 & 1 & 0 \\ \hline 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}.$$

The trellis of C_3 is shown in Figure 1. The code is extremely poor for a binary-symmetric channel, since the free Hamming distance is only 3. However, one of its cosets on the precoded $1 - D$ partial response channel can be described by the trellis in Figure 2. This code has free squared Euclidean distance equal to 14.

Figure 1. Trellis of C_3

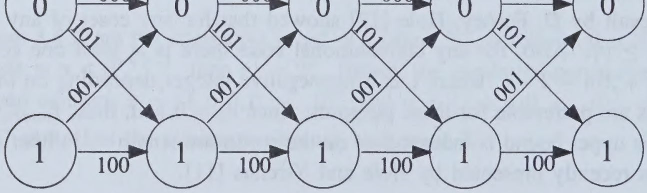
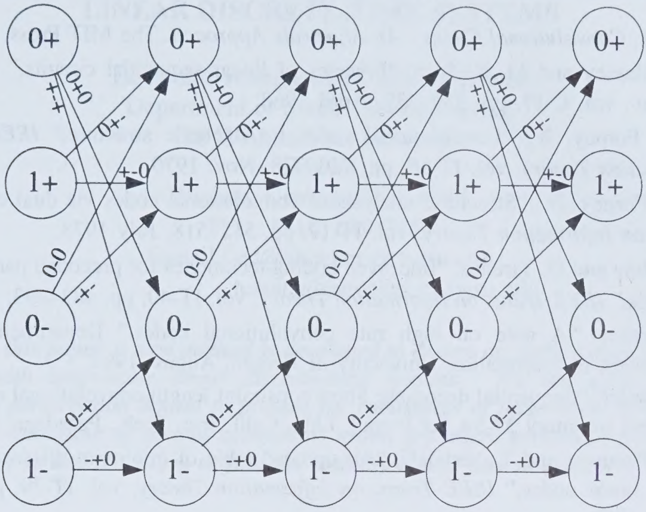


Figure 2. Trellis of $C_3 + (010)$ on precoded partial response channel



More codes and details can be found in [5, 13, 14].

4.3. Simplified decoding [10, 16]

Hole and Ytrehus [10] presented a decoding algorithm for "complex" convolutional codes, such as PUM codes. For many codes, the number of operations required by an ordinary Viterbi decoder is dominated by calculating the branch metrics of the best edges between the states of the convolutional code trellis. The basic idea of the new algorithm is to decode in two steps: In step 1, all the edges are efficiently calculated by a special *coset trellis*. In step 2, the decoding results of step 1 are applied to a simplified convolutional code trellis.

McEliece and Lin [17] recently described a decoding algorithm. As a particular example they showed that decoding of the code C_1 in Example 1 can be carried out with a total of 104 additions per information bit.

Ascetic codes are particularly suited for the decoding algorithm in [10]. The reason is that since few edge labels are used, each label is used more often. The code C_1 can be decoded with 141 operations (112 additions and 29 comparisons) per decoded bit. However, decoding of the more ascetic code C_2 requires only 64 operations (42 additions and 22 comparisons) per decoded information bit.

Hole and Hole [16] recently described the application of this decoding algorithm on the precoded $1 - D$ partial-response channel.

5. Conclusion

Ascetic codes have been presented. These codes are attractive for several applications. For many sets of parameters, these codes also have excellent Hamming or Euclidean distance properties.

References

[1] P. Piret, *Convolutional Codes - An Algebraic Approach*. The MIT Press, 1988.

[2] J. L. Massey and M. K. Sain, "Inverses of linear sequential circuits," *IEEE Trans. Comput.*, vol. C-17, pp. 330–337, April 1968.

[3] G. D. Forney, Jr., "Convolutional codes I: Algebraic structure," *IEEE Trans. on Information Theory*, vol. IT-16, pp. 720–738, Nov. 1970.

[4] G. D. Forney, Jr., "Structural analysis of convolutional codes via dual codes," *IEEE Trans. on Information Theory*, vol. IT-19, pp. 512–518, July 1973.

[5] K. J. Hole and Ø. Ytrehus, "Improved coding techniques for precoded partial-response channels," *IEEE Trans. on Information Theory*, vol. IT-40, pp. 482–493, March 1994.

[6] Ø. Ytrehus, "A note on high rate convolutional codes," Department report 68, Department of Informatics, University of Bergen, August 1992.

[7] J. A. Heller, "Sequential decoding: Short constraint length convolutional codes," space programs summary 37-54, Jet Propul. Lab., Calif. Inst. Tech., Pasadena, Dec. 1968.

[8] A. E. Brouwer and T. Verhoeff, "An updated table of minimum-distance bounds for binary linear codes," *IEEE Trans. on Information Theory*, vol. IT-39, pp. 662–677, March 1993.

[9] A. D. Wyner and R. B. Ash, "Analysis of recurrent codes," *IEEE Trans. on Information Theory*, vol. IT-9, pp. 143–156, 1963.

[10] M. F. Hole and Ø. Ytrehus, "Two-step trellis decoding of partial unit memory convolutional codes." Submitted for publication.

[11] K. J. Hole and Ø. Ytrehus, "Further results on cosets of convolutional codes with short maximum zero-run lengths," in *Proceedings of ISIT'95*, p. 146, September 1995.

[12] K. J. Hole, "Cosets of convolutional codes with short maximum zero-run lengths," *IEEE Trans. on Information Theory*, vol. IT-41, pp. 1145–1150, July 1995.

[13] K. J. Hole and Ø. Ytrehus, "Trellis codes for precoded partial-response channels: Further improved search techniques," in *Proceedings of ISITA'94*, pp. 475–479, November 1994.

[14] K. J. Hole, Ø. Ytrehus, R. Erstad, and M. F. Hole, "Convolutional codes for partial-response channels," in *Proceedings of Photonics East'95 Conference on Coding and Signal Processing for Information Storage*, 1995.

[15] J. K. Wolf and G. Ungerboeck, "Trellis coding for partial-response channels," *IEEE Trans. on Communication*, vol. COM-34, pp. 765–773, Aug. 1986.

[16] M. F. Hole and K. J. Hole, "Low complexity decoding of high-rate partial unit memory codes on precoded partial-response channels." Submitted for publication.

[17] R. J. McEliece and W. Lin, "The trellis complexity of convolutional codes," in *Proceedings of ISIT'95*, p. 131, September 1995.

OPTIMIZATION AND STABILITY ROBUSTNESS OF LARGE-SCALE LINEAR DISCRETE-TIME SYSTEMS

Ta-chung Wang and M. Edwin Sawan
Department of Electrical Engineering
The Wichita State University
Box 44
Wichita, Kansas 67260-0044
TEL : (316) 689-3415
E-mail : ed@shocker.ee.twsu.edu
tcwang@shocker.ee.twsu.edu

Abstract - In this paper, a new method is developed to design an optimization and stability robustness with large-scale linear discrete-time systems. We are going to use an unstructured perturbation bound to present for robustness of large-scale linear discrete-time systems consisting of several subsystems which are interconnected together, and we are also going to use $\mathcal{H}_\infty$ norm to find the stability bound and to compare with the new method.

I. INTRODUCTION

In this paper, a new method is developed to design optimization and stability robustness of large-scale linear discrete-time systems. The past work, in [1], presented an unstructured perturbation bound for robust stability of linear discrete-time systems, controller design, and standard Linear Quadratic (LQ) regulator. In this paper, we are going to continue the past work and to describe that how an unstructured perturbation bound for robust stability of linear discrete-time systems can be use for large-scale linear discrete-time case. The paper is also going to use another method for comparing with a new method which is going to be presented in this paper. In section II, a new method is going to be described for calculating an unstructured perturbation bound for robust stability of linear discrete-time systems using Lyapunov theory and singular value analysis. In section III, we are going to obtain the best technique for solving that when a linear discrete-time system consisting of several subsystems which are interconnected together. It is also going to describe that a linear discrete-time system can be stable for all the time when the system has many subsystems interconnected together and this main system is robustness. In section IV, we are going to offer an example to show that a linear time-invariant system is always stability robustness when it is connected with several subsystems which are interconnected together; and this section is also going to describe the bound for robust stability of linear discrete-time systems; and the section is going to make comparisons between the new method and $\mathcal{H}_\infty$ norm method.

II. STABILITY ROBUSTNESS BOUND FOR DISCRETE SYSTEMS

In this section, an unstructured perturbation bound is presented for robust stability of linear discrete-time systems which are given. The bound is derived using Lyapunov theory and singular analysis [2]. Consider the linear discrete-time systems described by the difference equation

$$x_i(k+1) = [A_i + \delta A_i(k)] x_i(k) \tag{1}$$

where $x_i(k)$ is the *n*-dimensional state vector, A_i is an $n \times n$ time-invariant asymptotically state matrix, and $\delta A_i(k)$ is an $n \times n$ time-varying uncertainty (perturbation) matrix [1].

The following theorem was presented in [2].

Theorem : The discrete-time system is stable if

$$\sigma_{\max}[\delta A_i(k)] < \mu_i \tag{2}$$

where
$$\mu_i = -\sigma_{\max}(A_i) + \sqrt{[\sigma_{\max}(A_i)]^2 + \frac{\sigma_{\min}(Q_i)}{\sigma_{\max}(P_i)}} \tag{3}$$

P_i is the positive definite solution of the discrete-time Lyapunov matrix equation,

$$A_i^T P_i A_i - P_i + Q = 0 \tag{4}$$

for any given positive definite symmetric Q_i, $\sigma_{\max}(.)$ and $\sigma_{\min}(.)$, represent the maximum and minimum singular values of (.). The singular values of any matrix θ are the symbol of $\sigma(\theta)$ and are defined by $\sigma(\theta) = [\lambda(\theta\theta^T)]^{\frac{1}{2}}$; where $\lambda(.)$ is the eigen value. The bound μ_i in (3) is maximum when $Q_i = I_i$ for which $\sigma_{\min}(Q_i) = 1_i$, as shown in [2].

III. PROBLEM STATEMENT

This section is going to assume an overall linear time-invariant system $x(k+1)=Ax(k)+Bu(k)$ comprises N subsystems which are interconnected together. The following equations is considers with the ith subsystems

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k) + z_i(k) \qquad x_i(0) = x_{i_0} \tag{5}$$

where $x_i(k)$ is the *n*-dimensional state vector, $u_i(k)$ is the *m*-dimensional control input vector, A_i is an $n \times n$ dimensional constant matrix, B_i is an $n \times m$ dimensional constant matrix, and $z_i(k)$ is the *q*-dimensional vector of interactions from the other subsystems, as

$$z_i(k) = \sum_{j \neq i}^{N} L_{ij} x_j(k) \qquad \begin{matrix} i = 1,2,\ldots,N \\ j = 1,2,\ldots,N \end{matrix} \tag{6}$$

where L_{ij} is an $q \times n$ dimensional constant matrix.

The problem is to design the control input $u_i(k)$ which minimizes the associated cost function

$$\min J = \frac{1}{2}\sum_{i=1}^{N}\sum_{k=0}^{N-1} x_i^T(k)Q_i x_i(k) + u_i^T(k)R_i u_i(k) \tag{7}$$

where Q_i and R_i are positive definite matrix with appropriate dimensions, $Q_i \geq 0$ and $R_i > 0$.

If A_i has uncertainty perturbation $\delta A_i(k)$, the equation (5) can be re-written as

$$x_i(k+1) = \left[A_i + \delta A_i(k)\right]x_i(k) + B_i u_i(k) + z_i(k) \tag{8}$$

then using the above equation obtain the boundary of $\delta A_i(k)$ for system be stable.

The following step is to use the Lagrangian as

$$\begin{aligned}\mathcal{L} = \sum_{i=1}^{N}\sum_{k=0}^{N-1} \{ & \frac{1}{2}\left[x_i^T(k)Q_i x_i(k) + u_i^T(k)R_i u_i(k)\right] \\ & + \alpha_i^T(k)\left[-z_i(k) + \sum_{j\neq i}^{N} L_{ij}x_j(k)\right] \\ & + \beta_i^T(k+1)\left[-x_i(k+1) + A_i x_i(k) + B_i u_i(k) + z_i(k)\right]\}\end{aligned} \tag{9}$$

where $\alpha_i(k)$ and $\beta_i(k)$ are the Lagrange multiplier matrices which are positive definite.

$$\sum_{i=1}^{N}\sum_{j\neq i}^{N} \alpha_i^T(k)L_{ij}x_j(k) = \sum_{j=1}^{N}\sum_{i\neq j}^{N} \alpha_j^T(k)L_{ji}x_i(k) \tag{10}$$

Set

$$\begin{aligned}\mathcal{L} = & \sum_{i=1}^{N}\sum_{k=0}^{N-1} G_i \\ = & \sum_{i=1}^{N}\sum_{k=0}^{N-1} \{\frac{1}{2}\left[x_i^T(k)Q_i x_i(k) + u_i^T(k)R_i u_i(k)\right] \\ & - \alpha_i^T(k)z_i(k) + \sum_{i\neq j}^{N} \alpha_j^T(k)L_{ji}x_i(k) \\ & + \beta_i^T(k+1)\left[-x_i(k+1) + A_i x_i(k) + B_i u_i(k) + z_i(k)\right]\}\end{aligned} \tag{11}$$

The notice from the equation (11) is that $\mathcal{L}$ is additively separable for any given $z_i(k)$ and $\alpha_i(k)$ trajectories. It implies that if $z_i(k)$ and $\alpha_i(k)$ are fixed or supplied by a second-level unit, the minimization of $\mathcal{L}$ can be carried out by N independent minimization, the i^{th} of which is G_i in equation (11). This leaves the problem of improving $z_i(k)$ and $\alpha_i(k)$ such that the overall optimum is achieved.

The matrix gradient method is used to obtain the following necessary conditions :

$$\frac{\partial G_i}{\partial \beta_i(k+1)} = 0 \tag{12a}$$

$$\Rightarrow \quad x_i(k+1) = A_i x_i(k) + B_i u_i(k) + z_i(k)$$

$$\frac{\partial G_i}{\partial x_i(k)} = \beta_i(k)$$

$$\Rightarrow \beta_i(k) = Q_i x_i(k) + A_i^T \beta_i(k+1) + \sum_{i \neq j}^{N} L_{ji}^T \alpha_j(k) \tag{12b}$$

$$\beta_i(N) = 0_i$$

$$\frac{\partial G_i}{\partial u_i(k)} = 0 \tag{12c}$$

$$\Rightarrow \quad u_i(k) = -R_i^{-1} B_i^T \beta_i(k+1)$$

$$\frac{\partial G_i}{\partial \alpha_i(k)} = 0 \tag{12d}$$

$$\Rightarrow \quad z_i(k) = \sum_{j \neq i}^{N} L_{ij} x_j(k)$$

The necessary conditions on the above constitute the first-level, at the second level, the sequences $z_i(k)$ and $\alpha_i(k)$ are improved by prediction rule [3],

$$\begin{bmatrix} \alpha_i(k) \\ z_i(k) \end{bmatrix}_{m+1} = \begin{bmatrix} \beta_i(k) \\ \sum_{j \neq i}^{N} L_{ij} x_j(k) \end{bmatrix}_m$$

where *m* is the iteration index. This method is called the interaction prediction method.

Let

$$\beta_i(k) = K_i(k) x_i(k) + s_i(k) \tag{13}$$

where $s_i(k)$ is the open-loop compensation vector and $K_i(k)$ is the solution of the Riccati equation.

The next step is to substitute equation (13) into equation (12c),

$$u_i(k) = -R_i^{-1} B_i^T \left[K_i(k+1) x_i(k+1) + s_i(k+1) \right] \tag{14}$$

The following is to substitute equation (14) into equation (12a),

$$\begin{aligned} x_i(k+1) &= A_i x_i(k) + B_i \left\{ -R_i^{-1} B_i^T \left[K_i(k+1) x_i(k+1) + s_i(k+1) \right] \right\} \\ &\quad + z_i(k) \\ &= A_i x_i(k) - B_i R_i^{-1} B_i^T K_i(k+1) x_i(k+1) \\ &\quad - B_i R_i^{-1} B_i^T s_i(k+1) + z_i(k) \end{aligned}$$

$$\begin{aligned}\Rightarrow x_i(k+1) = &\left[I_i + B_i R_i^{-1} B_i^{-1} K_i(k+1)\right]^{-1} \\ &\left[A_i x_i(k) + z_i(k) - B_i R_i^{-1} B_i^T s_i(k+1)\right]\end{aligned} \tag{15}$$

The following is to manipulate equation (12b) using equation (13) and (15),

$$\begin{aligned}K_i(k)x_i(k) + s_i(k) = &\ Q_i x_i(k) \\ &+A_i^T\left[K_i(k+1)x_i(k+1) + s_i(k+1)\right] \\ &-\sum_{i \neq j}^{N} L_{ji}^T \alpha_j(k) \\ = &\ Q_i x_i(k) \\ &+A_i^T\{\left[I_i + B_i R_i^{-1} B_i^{-1} K_i(k+1)\right]^{-1} \\ &\qquad \left[A_i x_i(k) + z_i(k) - B_i R_i^{-1} B_i^T s_i(k+1)\right] \\ &\qquad +s_i(k+1)\} \\ &-\sum_{i \neq j}^{N} L_{ji}^T \alpha_j(k)\end{aligned}$$

$$\begin{aligned}\Rightarrow\ &\{K_i(k) - Q_i - A_i^T K_i(k+1)[I_i + B_i R_i^{-1} B_i^T K_i(k+1)]^{-1} A_i\} x_i(k) \\ &+s_i(k) - A_i^T s_i(k+1) \\ &+A_i^T K_i(k+1)[I_i + B_i R_i^{-1} B_i^T K_i(k+1)]^{-1} B_i R_i^{-1} B_i^T s_i(k+1) \\ &-A_i^T K_i(k+1)[I_i + B_i R_i^{-1} B_i^T K_i(k+1)]^{-1} z_i(k) \\ &-\sum_{i \neq j}^{N} L_{ji}^T \alpha_j(k) \\ &= 0\end{aligned} \tag{16}$$

Since equation (16) is true for arbitrary $x_i(k)$, thus

$$\begin{aligned}K_i(k) = &\ Q_i - A_i^T K_i(k+1)[I_i + B_i R_i^{-1} B_i^T K_i(k+1)]^{-1} A_i \\ &K_i(N) = 0\end{aligned} \tag{17}$$

$$\begin{aligned}s_i(k) = &\ A_i^T\{I_i - K_i(k+1)[I_i + B_i R_i^{-1} B_i^T K_i(k+1)]^{-1} B_i R_i^{-1} B_i^T\} s_i(k+1) \\ &+ A_i^T K_i(k+1)[I_i + B_i R_i^{-1} B_i^T K_i(k+1)]^{-1} z_i(k) \\ &+ \sum_{i \neq j}^{N} L_{ji}^T \alpha_j(k) \\ &s_i(N) = 0\end{aligned} \tag{18}$$

The matrix inversion Lemma is used in equation (17) and (18),

$$\begin{aligned} K_i(k) &= Q_i + A_i^T K_i(k+1) A_i \\ &\quad - A_i^T K_i(k+1) B_i [R_i + B_i^T K_i(k+1) B_i]^{-1} B_i^T K_i(k+1) A_i \end{aligned} \tag{19}$$

$$\begin{aligned} s_i(k) &= A_i^T [I_i + K_i(k+1) B_i R_i^{-1} B_i^T]^{-1} s_i(k+1) \\ &\quad + A_i^T K_i(k+1)[I_i + B_i R_i^{-1} B_i^T K_i(k+1)]^{-1} \\ &\quad + \sum_{i\neq j}^{N} L_{ji}^T \alpha_j(k) \end{aligned} \tag{20}$$

The equation (20) can be re-written as

$$\begin{aligned} s_i(k+1) &= V_i^{-1} s_i(k) - V_i^{-1} W_i \sum_{j\neq i}^{N} L_{ij} x_j(k) \\ &\quad - V_i^{-1} \sum_{i\neq j}^{N} L_{ji} [K_j(k) x_j(k) + s_j(k)] \end{aligned} \tag{21}$$

where

$$V_i = A_i^T [I_i + K_i(k+1) B_i R_i^{-1} B_i^T]^{-1}$$

$$W_i = A_i^T K_i(k+1)[I_i + B_i R_i^{-1} B_i^T K_i(k+1)]^{-1}$$

The following design is going to set that $\underline{x}$ (k) is the overall state vector of this system, $\underline{u}$ (k) is the overall control input vector of this system, $\underline{s}$ (k) is the overall open-loop part of the compensator, and A, B, L, Q, R, and K are the constant matrices for this overall system.

The equation (21) can be re-written for the integrated system as

$$\underline{s}(k+1) = C\underline{s}(k) - D\underline{x}(k) \tag{22}$$

where

$$C = \left(V^{-1} - V^{-1} L^T\right)$$

$$D = \left(V^{-1} W L + V^{-1} L^T K\right)$$

$$V = \sum_{i=1}^{N} V_i$$

$$W = \sum_{i=1}^{N} W_i$$

$$L = \sum_{j\neq i}^{N} L_{ij}$$

$$K = \sum_{i=1}^{N} K_i$$

Now, the next step is going to present the transition matrix, as following

$$\underline{s}(k) = \phi_1(k,0)\underline{s}(0) - \sum_{\ell=0}^{k-1} \phi_1(k,\ell+1) D \underline{x}(\ell) \tag{23}$$

where $\phi_1(k,0)$ is the transition matrix associated with ($V^{-1} - V^{-1}L^T$).

The following step considers with the composite system

$$\underline{x}(k+1) = A\underline{x}(k) + B\underline{u}(k) \tag{24}$$

and the cost function

$$J = \frac{1}{2}\sum_{k=0}^{N-1} \underline{x}^{T}(k)Q\underline{x}(k) + \underline{u}^{T}(k)R\underline{u}(k) \tag{25}$$

The optimal control $\underline{u}$ (k) is

$$\underline{u}^{*}(k) = -\left[R + B^{T}P(k+1)B\right]^{-1} B^{T}P(k+1)A\underline{x}(k) \tag{26}$$

where P(k) is the solution of the composite Riccati equation.
From the above equations, the equation (24) becomes

$$\begin{gathered}\underline{x}(k+1) = \left\{A - B\left[R + B^{T}P(k+1)B\right]^{-1} B^{T}P(k+1)A\right\}\underline{x}(k) \\ \underline{x}(k) = \phi_{2}(k,0)\underline{x}(k)\end{gathered} \tag{27}$$

where $\phi_2(k,0)$ is the transition matrix associated with

$$\left\{A - B\left[R + B^{T}P(k+1)B\right]^{-1} B^{T}P(k+1)A\right\}$$

The equation (23) can be re-written by substituting $\underline{x}$ (k) from the equation (27) as

$$\underline{s}(k) = \phi_{1}(k,0)\underline{s}(0) - \sum_{\ell=0}^{N-1} \phi_{1}(k,\ell+1)D\phi_{2}(\ell,0)\underline{x}(0) \tag{28}$$

The final condition, $\underline{s}$ (N) = 0, is used for the equation (28), the following is obtained

$$0 = \phi_{1}(N,0)\underline{s}(0) - \sum_{\ell=0}^{N-1} \phi_{1}(N,\ell+1)D\phi_{2}(\ell,0)\underline{x}(0) \tag{29}$$

From the equation (29), the following is obtained

$$\begin{aligned}\underline{s}(0) &= \phi_{1}(0,N)\left\{\sum_{\ell=0}^{N-1} \phi_{1}(N,\ell+1)D\phi_{2}(\ell,0)\right\}\underline{x}(0) \\ &= E(N,0)\underline{x}(0)\end{aligned} \tag{30}$$

where

$$E(N,0) = \sum_{\ell=0}^{N-1} \phi_{1}(0,\ell+1)D\phi_{2}(\ell,0) \tag{31}$$

In general,

$$\underline{s}(k) = E(N,k)\underline{x}(k) \tag{32}$$

From the equation (32), the equation (31) can be re-written as

$$E(N,k) = \sum_{\ell=0}^{N-1} \phi_1(k,\ell+1)D\phi_2(\ell,k) \tag{33}$$

The open-loop compensation vector $\underline{s}$ (k) and the state vector $\underline{x}$ (k) are related by a linear transformation; when N→ ∞, K(k+1) = K(k) = K and P(k+1) = P(k) = P are constant matrices. Consequently, E(N,k) becomes a constant matrix, which indicates that for the infinite-time regulator, and is time-invariant

$$\begin{aligned} \underline{s}(k) &= E\underline{x}(k) \\ \Rightarrow\quad E &= s\underline{x}^{-1}(k) \end{aligned} \tag{34}$$

An alternative procedure would be to solve the problem off-line N times successively from the initial conditions.

$$\underline{x}(0) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \cdots \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

Then, E = s = [$\underline{s}_1$, $\underline{s}_2$, . . . , $\underline{s}_N$]; here, the compensation is off-line and in a decentralized way.

From the equation (14) and (34), the following obtains the overall closed-loop control

$$\begin{aligned} \underline{u}(k) &= -R^{-1}B^T\left[K + E\right]\underline{x}(k) \\ &= -R^{-1}B^TK\underline{x}(k) - R^{-1}B^TE\underline{x}(k) \end{aligned} \tag{35}$$

where $R^{-1}B^TK$ is block diagonal matrix with $R_i^{-1}B_i^TK_i$ as the block elements, $R^{-1}B^TK$ represents the off-diagonal elements in the feedback gain matrix.

In the equation (19), obtain K_i is independent on the initial condition. The result will be the feedback control, the equation (35), which is independent on the initial condition; thus, the equation (35) can bring the discrete system back to steady state optimally from any initial conditions.

So, from the equation (35), the optimal control can be re-written as

$$\begin{gathered} u_i(k) = -R_i^{-1}B_i^TK_ix_i(k) - R_i^{-1}B_i^TEx_j(k) \\ i \neq j \end{gathered} \tag{36}$$

Then, substituting the equation (36) into the equation (1) and (5), the following obtain,

$$
\begin{aligned}
&x_i(k+1) = A_{c_i}\underline{x}(k) \\
&\Rightarrow\ x_i(k+1) = \left[A_{c_i} + \delta A_i(k)\right]\underline{x}(k)
\end{aligned}
\tag{37}
$$

From the equation (25), the optimal cost can be obtained as

$$
\begin{aligned}
J^* = \frac{1}{2}\Bigg\{\sum_{k=0}^{N-1}\Big\{&\underline{x}^T(k)Q\underline{x}(k) \\
&+\left[-\left[R+B^TP(k+1)B\right]^{-1}B^TP(k+1)A\underline{x}(k)\right]^T R \\
&\left[-\left[R+B^TP(k+1)B\right]^{-1}B^TP(k+1)A\underline{x}(k)\right]\Big\}\Bigg\}
\end{aligned}
\tag{38}
$$

IV. EXAMPLE

Now, we are going to present an example, as following,

$$
\begin{gathered}
N = 3,\quad i = 1,2,\ldots,N,\quad j = 1,2,\ldots,N,\quad i \neq j \\
A_1 = 1 \quad B_1 = 1 \quad R_1 = 1 \quad Q_1 = 1 \\
A_2 = 1 \quad B_2 = 1 \quad R_2 = 1 \quad Q_2 = 1 \\
A_3 = 1 \quad B_3 = 1 \quad R_3 = 1 \quad Q_3 = 1 \\
C = I \\
L_{ij} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad x_o = \begin{bmatrix} x_{o_1} \\ x_{o_2} \\ x_{o_3} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
x(k+1) = \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} \\
\Rightarrow x(k+1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} u(k) + G\omega(k) + z_i(k) \\
y(k) = Cx(k)
\end{gathered}
$$

Then, we can obtain the optimal cost, J* = 20.1537, and the unstructured perturbation bound, $\mu \approx 0.1521$; from [3], we can calculate and find the $\mathcal{H}_\infty$-norm bound, $\gamma \approx 0.1906$.

V. CONCLUSION

There are many methods to solve the linear discrete-time systems consisting of an interconnection of subsystems; only one method is considered in this paper. The basic idea for this method is illustrated with an example. The interaction prediction method requires very little second-level calculation. If one wants to calculate by hand, it is also very complicated.

The best way to solve this kind of problem is by using the computer for the tool. Although the method shown in this paper is very complicated and needs computer to be a tool, it is the best technique to solve the robustness of the linear discrete-time system and to prove it is always stable in all cases. And from the example, we also can know that using the new method to calculate the stability bound is better than using $\mathcal{H}_\infty$-norm to calculate.

REFERENCES

[1] Sri R. Kolla and James B. Farison, "Reduced-Order Dynamic Compensator Design for Stability Robustness of Linear Discrete-Time Systems," *IEEE Trans. Automat. Contr.*, vol. 36, pp. 1077-1081, Sept. 1991.

[2] R. Kolla, R. K. Yedavalli, and J. B. Farison, "Robust stability bounds on time-varying perturbations for state-space models of linear discrete-time systems," *Int. J. Contr.*, vol. 50, pp. 151-159, 1989.

[3] R. A. PAZ and J. V. MEDANIC, "$\mathcal{H}_\infty$ control in discrete time : state feedback control and norm bounds," *International Journal of Control.*, vol. 55, pp. 1405-1424, 1992.

[4] Magdi S. Mahmoud, Mohamed F. Hassan, and Mohamed G. Darwish, "***Large-Scale Control Systems Theories and Techniques*** ", Marcel Dekker, New York, 1985.

[5] Hassan, M. F. R. Hurteau, M. G. Singh, and A. Titli, A Stochastic Hierarchical Control of a Large-Scale River System, "***Proc. IFAC Seventh World Congr.*** ", Helsinki, June 12-16, 1978.

[6] Spyros G. Tzafestas, "***Multivariable Control*** ", Reidel, Holland, 1984.

[7] T. Ishihara, "Robust stability bounds for a class of discrete-time regulators with computational delays," in *Proc. 26th Allerton Conf. Comm., Contr. and Comp.*, Monticello, IL, Sept. 28-30, 1988, pp. 893-902.

[8] P. Dorato, Ed., *Robust Control.* New York: IEEE Press, 1987.

[9] Lan R. Petersen, "Complete Results for a Class of State Feedback Disturbance Attenuation Problems," *IEEE Trans. Automat. Contr.*, vol. 34, pp. 1196-1199, Nov. 1989.

[10] John C. Doyle, Keith Glover, Pramod P. Khargonekar, and Bruce A. Francis, "State-Space Solutions to $\mathcal{H}_2$ and $\mathcal{H}_\infty$ Control Problems," *IEEE Trans. Automat. Contr.*, vol. 34, pp. 831-847, Aug. 1989.

[11] Canes and S. Lafortune, "Adaptive control with recursive identification for stochastic linear systems," *IEEE Trans. Automat. Contr.*, vol. AC-29, pp. 312-321, Apr. 1984.

Design of Optimal Sampled-Data Repetitive Control Systems

Ali Langari **Bruce A. Francis**
ali@control.utoronto.ca **francis@control.utoronto.ca**

Systems Control Group
Electrical and Computer Engineering
University of Toronto
Toronto, Canada, M5S 1A4

Abstract

A design procedure for optimal sampled-data repetitive control systems is presented. The design is based on the minimization of the power-norm of the steady-state tracking error, which takes the intersample behavior into consideration. State-space formulas are obtained for the controller. This technique is demonstrated by an example.

1 Introduction

Repetitive control systems are those in which the controller is designed to make a plant track or reject unknown periodic reference commands or disturbance signals of which only the period T is known. These systems, introduced by Inoue et al. [9], have found many applications in control engineering, such as rejection of power supply interferences [13], accurate control of disk-drive systems [3], control of robotic manipulators which repeatedly perform the same task [14], and control of peristaltic pumps [8]. Theoretically, these systems have been studied in continuous time by Hara et al. [7] and Yamamoto [18], in discrete time by Tomizuka et al. [16], and also in a sampled-data framework by the authors [11], where the issue of the intersample behavior is addressed.

In this paper, we present a new formulation for repetitive control systems which focuses on an issue overlooked in the previous research works. So far the attempt in the design of repetitive controllers has been based on the assumption that the periodic signals are rich in each and every single harmonic of the fundamental frequency. This is not a very realistic assumption and it results in complex infinite-dimensional controllers in continuous time, possible poor intersample behavior in discrete-time designs, and conservative designs in a sampled-data formulation. In short, the previous design techniques tend to overdesign.

To remedy this, we make the observation that in practical situations, the periodic reference commands or disturbance signals are not so rich in their frequency content and they usually have only a few harmonics. In our first attempt, we make this even simpler by assuming that the periodic signals are *known*; that is, we assume they are fixed. While the internal model principle makes it very simple to find an answer to this problem in continuous time, the solution in a sampled-data framework, where the continuous-time plant is in connection with the discrete-time controller through sample and hold devices, is not straightforward.

Our approach to this problem is to minimize the power of the steady-state tracking error. The main feature of this formulation is that it puts a major emphasis on the intersample error. The intersample error in systems subject to periodic inputs has been previously referred to as ripple in some papers. In [5], Franklin et al. proposed a technique for ripple-free sampled-data servomechanisms. But they required that the continuous-time part of the system include an internal model for the periodic inputs. This is a strong condition and in case that the continuous-time plant does not have the required internal model, some analog pre-compensation is necessary. Hara et al. in [6] proposed a different technique for ripple attenuation in repetitive control systems which involves generalized hold functions and asynchronous sampling. Compared to the standard sampled-data design, where the hold is zero-order and is synchronous with the sampler, this suffers from more complexity. For a discussion on the ability of sampled-data systems in tracking various command inputs see [2].

To formulate our problem, we need the following definitions. A continuous-time signal $w : \mathbb{R}_+ \to \mathbb{C}^p$ is called *periodic of period* $T > 0$ if it satisfies $w(t+T) = w(t), \ \ t \geq 0$. Also define

$$\mathcal{W}_T = \{w : w \text{ is of period T and } \frac{1}{T}\int_0^T w(t)^* w(t) dt < \infty\},$$

where star denotes conjugate transpose. For $w \in \mathcal{W}_T$ define its power-norm by

$$\|w\| = \left[\frac{1}{T}\int_0^T w(t)^* w(t) dt\right]^{\frac{1}{2}}.$$

Analogously, a discrete-time signal $\omega : \mathbb{Z}_+ \to \mathbb{C}^p$ satisfying $\omega(n+m) = \omega(n), \ \ n \geq 0$, is *periodic of period* m and its *power-norm* is

$$\|\omega\| = \left[\frac{1}{m}\sum_{n=0}^{m-1} \omega(n)'\omega(n)\right]^{1/2}.$$

Finally, define

$$\Omega_m = \{\omega : \omega \text{ is of period } m\}.$$

2 Problem Statement

The setup of interest is shown in Figure 1 where solid and dotted lines represent continuous-time and discrete-time signals, respectively. The plant G is a continuous-time finite-dimensional linear time-invariant (FDLTI) system with a minimal realization of the form

$$\hat{g}(s) = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & 0 & 0 \end{array}\right].$$

The input w contains all reference/disturbance signals, which is assumed to be a *known* periodic signal of period T, and the output z contains all the tracking errors to be minimized. The controller K is a sampled-data controller with two assumptions regarding

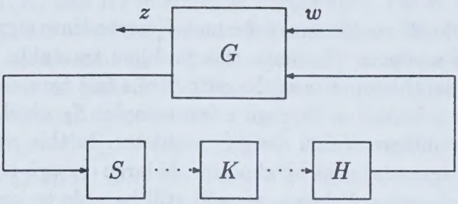


Figure 1: The sampled-data repetitive control system.

its sampling period h: (1) There is an integer number of sampling periods within one period of the periodic input, i.e., $Mh = T$ for some integer M, and (2) the matrix A in the realization of G satisfies the nonpathological sampling condition that for each eigenvalue λ of A with $Re\lambda \geq 0$, none of the points $\lambda + j\frac{2\pi k}{h}$, $k \neq 0$, is an eigenvalue of A. This condition guarantees stabilizability of the system. Bring in Figure 2, the discretization of the system in Figure 1. The class of controllers over which the optimization is performed is $\mathcal{C}$, the class of LTI controllers for which the A-matrix of the discretized system in Figure 2 is stable. As shown in [1], under mild technical conditions these controllers achieve internal stability for the sampled-data setup.

The system in Figure 1 is *not* a LTI system; however, it exhibits some of the nice properties that appear in LTI signals and systems theory. The following useful result counts for one that we will use in our problem formulation [12]:

Theorem 1 *Suppose* $K \in \mathcal{C}$. *Then in the sampled-data setup of Figure 1, for each* $w \in \mathcal{W}_T$, *there exist unique signals* z_{ss} *in* $\mathcal{W}_T$ *and* z_{tr} *in* $\mathcal{L}_2(\mathrm{R}_+)$ *such that* $z = z_{ss} + z_{tr}$.

This result shows that for a stabilizing controller the output z approaches a steady-state periodic component for any periodic input. For a given periodic input, the smaller this steady-state component, the closer we are to the goal of the design, that is, tracking or rejection of reference or disturbance signals. That is why we state the *optimal sampled-data repetitive control problem* as

$$\min_{K\in\mathcal{C}} \|z_{ss}\|. \tag{1}$$

A very important feature of this minimization problem is that since z_{ss} is a continuous-time signal, it incorporates the intersample error directly into the design.

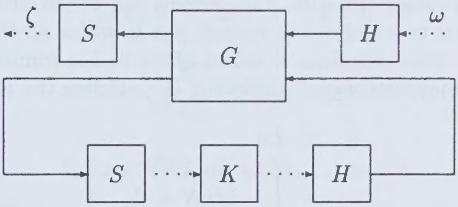


Figure 2: Discretization of the sampled-data system.

Note in Figure 1 that there are two time periods: T is the period of the periodic input w; h is the sampling period. By assumption, $T = Mh$, where M is an integer.

Problem Reformulation via Fast Sampling

Problem (1) involves both continuous-time and discrete-time signals and hence is not amenable to a simple analysis. To make this problem tractable, following Keller and Anderson [10], we model the input w as the output of a fast zero-order hold H_f of period h/N and the output z is looked at through a fast sampler S_f which is synchronized with H_f; see Figure 3. The integer N is a design parameter. In this way, the periodic input w may not be the one intended, but by choosing N large enough it can be made as close as desired. Also, by choosing N large, we will still be able to control the intersample behavior of the output. So, it is expected that the setup of Figure 3 will emulate the sampled-data setup of Figure 1 for large N. The advantage of this new system, which is a pure discrete-time system, is that the discrete-time counterpart of problem (1) defined for this setup has a state-space solution.

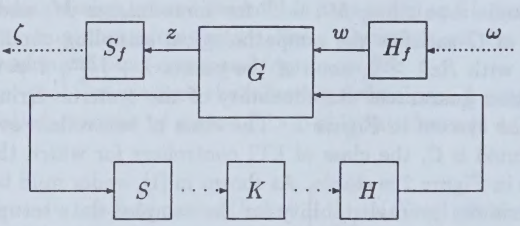


Figure 3: Fast discretization to emulate Figure 1.

Note that Figure 3 has three time periods: T and h as before, and h/N, the period of the fast sampler. Thus $T = Mh = MN(h/N)$. Hence if ω is periodic of period MN, then w is periodic of period T.

Theorem 2 *In Figure 3, suppose $K \in \mathcal{C}$. For each periodic ω of period MN, there exist unique signals ζ_{ss} in Ω_{MN} and ζ_{tr} in $\ell_2(\mathbb{Z}_+)$ such that $\zeta = \zeta_{ss} + \zeta_{tr}$.*

The new problem can then be stated as

$$\min_{K\in\mathcal{C}} \|\zeta_{ss}\|. \tag{2}$$

In Figure 3 there are discrete-time signals of two rates; some of them are referred to the base period h on the real-time clock while the others are referred to the subperiod h/N of the base period. In such a situation, the analysis can be performed more simply by a technique that converts to a single-rate system, one in which all the signals are referred to the base period h. This technique is called *lifting* and is commonly used (under the name "blocking") in multirate signal processing [17]. Define the *lifting operator* L by

$$\begin{aligned} \underline{v} &= Lv \\ \underline{v}(k) &= \begin{bmatrix} v(kN) \\ v(kN+1) \\ \vdots \\ v(kN+N-1) \end{bmatrix}. \end{aligned}$$

So, if v is a signal which is referred to subperiod h/N, its *lifted* associate $\underline{v}$ can be referred to period h. Note though that the dimension of the lifted signal is N times more.

Introduce the lifting operator and its inverse in Figure 3 and let P denote the system comprised of G, S, H, S_f and H_f to arrive at Figure 4(a). For $K \in \mathcal{C}$, Theorem 2 allows a decomposition of ζ into two components for $\omega \in \Omega_{MN}$, its steady state component ζ_{ss} and its transient component ζ_{tr}. Also L obviously maps periodic signals of period MN to periodic signals of period M. Since $\underline{\zeta} = L\zeta$ and $\underline{\omega} = L\omega$, for a stabilizing controller, there is a natural decomposition for $\underline{\zeta}$ when $\underline{\omega} \in \Omega_M$, namely, $\underline{\zeta} = \underline{\zeta}_{ss} + \underline{\zeta}_{tr}$ where $\underline{\zeta}_{ss} := L\zeta_{ss} \in \Omega_M$ and $\underline{\zeta}_{tr} := L\zeta_{tr} \in \ell_2(\mathbb{Z}_+)$. It is straightforward to show that the power-norm is scaled up under L by the factor $\sqrt{N}$, that is, for $\underline{v} = Lv$, where $v \in \Omega_{MN}$, we have $\|\underline{v}\| = \sqrt{N}\|v\|$. Therefore, minimization problem (2) can be stated in terms of the lifted output $\underline{\zeta}_{ss}$

$$\min_{K \in \mathcal{C}} \|\underline{\zeta}_{ss}\|. \tag{3}$$

The advantage of (3) is that if the lifting operators are taken into P, the resulting system in Figure 4(b) with

$$\underline{P} := \begin{bmatrix} L & 0 \\ 0 & I \end{bmatrix} P \begin{bmatrix} L^{-1} & 0 \\ 0 & I \end{bmatrix} \tag{4}$$

is not only single-rate and linear but also time invariant (for the state space representation, see [2]). This allows us to employ frequency-domain formulas which help to obtain a closed-form solution for an optimal controller. This is discussed in the next section.

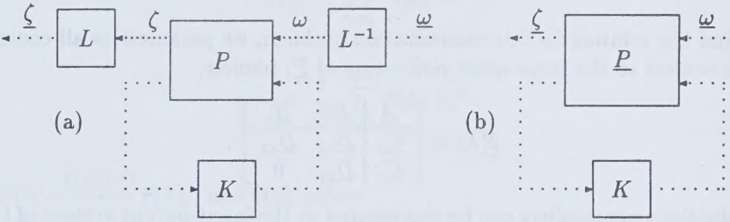


Figure 4: (a) Lifting. (b) Single-rate system.

3 Design Procedure

In this section we present the solution to (3). We start with some preliminaries. For $\omega(n) \in \Omega_m$, the standard discrete Fourier transform (DFT) equations [15] are defined by

$$\tilde{\omega}(k) = \sum_{n=0}^{m-1} \omega(n)\, \lambda_n^k\,, \ k \geq 0 \tag{5}$$

$$\omega(n) = \frac{1}{m} \sum_{k=0}^{m-1} \tilde{\omega}(k)\, \lambda_k^{-n}\,, \ \ n \geq 0, \tag{6}$$

where

$$\lambda_k := e^{-j\frac{2\pi}{m}k}\,, \ \ k = 0, 1, \ldots, m-1. \tag{7}$$

Symbolically, we have

$$\tilde{\omega}(k) \xleftarrow{DFT} \omega(n)$$

$$\omega(n) \xleftarrow{DFT^{-1}} \tilde{\omega}(k)\ .$$

The follwing useful lemma connects the power of a periodic discrete-time signal to the ℓ_2-norm of its DFT.

Lemma 1 *For* $\omega(n) \in \Omega_m$

$$\|\omega\| = \left[\frac{1}{m}\sum_{k=0}^{m-1} \tilde{\omega}(k)^* \, \tilde{\omega}(k)\right]^{1/2} = \left[\frac{1}{m}\sum_{k=0}^{m-1} \|\tilde{\omega}(k)\|^2\right]^{1/2}. \tag{8}$$

So (3) can be written as

$$\min_{K\in\mathcal{C}} \frac{1}{M}\sum_{k=0}^{M-1} \left\|\tilde{\underline{\zeta}}_{ss}(k)\right\|^2 . \tag{9}$$

DFTs of $\underline{\zeta}_{ss}$ and $\underline{\omega}$ are related in a simple way. In Figure 4(b), denote the mapping $\underline{\omega} \mapsto \underline{\zeta}$ by $\underline{\Phi}_K$ with transfer function $\hat{\underline{\phi}}_K(\lambda)$ ($\lambda := z^{-1}$). For $K \in \mathcal{C}$, this mapping defines a stable LTI discrete-time system and from [12] we have

$$\tilde{\underline{\zeta}}_{ss}(k) = \hat{\underline{\phi}}_K(\lambda_k)\, \tilde{\underline{\omega}}(k), \quad k = 1, 2, \ldots, M-1.$$

So the minimization problem (9) turns into

$$\min_{K\in\mathcal{C}} \frac{1}{M}\sum_{k=0}^{M-1} \left\|\hat{\underline{\phi}}_K(\lambda_k)\, \tilde{\underline{\omega}}(k)\right\|^2 . \tag{10}$$

To find the solution to this minimization problem, we parametrize all controllers in $\mathcal{C}$. With respect to the state-space realization of $\underline{P}$, namely,

$$\hat{\underline{p}}(\lambda) = \left[\begin{array}{c|cc} \underline{A} & \underline{B}_1 & \underline{B}_2 \\ \hline \underline{C}_1 & \underline{D}_{11} & \underline{D}_{12} \\ \underline{C}_2 & \underline{D}_{21} & 0 \end{array}\right], \tag{11}$$

all the stabilizing controllers can be represented as the input-output system of the block diagram in Figure 5 [4], where Q is stable FDLTI and J has the realization

$$\hat{\jmath}(\lambda) = \left[\begin{array}{c|cc} \underline{A} + \underline{B}_2 F + H\underline{C}_2 & -H & \underline{B}_2 \\ \hline F & 0 & -I \\ -\underline{C}_2 & I & 0 \end{array}\right] \tag{12}$$

in which F and H are such that $\underline{A} + \underline{B}_2 F$ and $\underline{A} + H\underline{C}_2$ are stable.

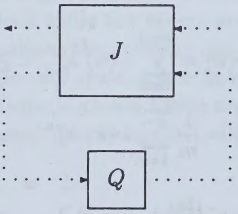


Figure 5: Stabilizing controllers

The advantage of this characterization is that the input-output system in Figure 4(b) from $\underline{\omega}$ to $\underline{\zeta}$ is a linear fractional transformation of Q, that is,

$$\hat{\underline{\phi}}_K(\lambda) = \hat{\underline{\phi}}_1(\lambda) + \hat{\underline{\phi}}_2(\lambda)\, \hat{q}(\lambda)\, \hat{\underline{\phi}}_3(\lambda), \tag{13}$$

where

$$\begin{aligned}
\hat{\underline{\phi}}_1(\lambda) &= \left[\begin{array}{cc|c} \underline{A}+\underline{B}_2F & \underline{B}_2F & \underline{B}_1 \\ 0 & \underline{A}+H\underline{C}_2 & -\underline{B}_1-H\underline{D}_{12} \\ \hline \underline{C}_1+\underline{D}_{12}F & \underline{D}_{12}F & \underline{D}_{11} \end{array}\right] \\
\hat{\underline{\phi}}_2(\lambda) &= \left[\begin{array}{c|c} \underline{A}+\underline{B}_2F & \underline{B}_2 \\ \hline \underline{C}_1+\underline{D}_{12}F & \underline{D}_{12} \end{array}\right] \\
\hat{\underline{\phi}}_3(\lambda) &= \left[\begin{array}{c|c} \underline{A}+H\underline{C}_2 & \underline{B}_1+H\underline{D}_{12} \\ \hline \underline{C}_2 & \underline{D}_{21} \end{array}\right].
\end{aligned} \tag{14}$$

Then the sampled-data repetitive control problem becomes

$$\min_{\hat{q}(\lambda)\ stable} \frac{1}{M}\sum_{k=0}^{M-1} \left\|\hat{\underline{\phi}}_1(\lambda_k)\underline{\tilde{\omega}}(k)+\hat{\underline{\phi}}_2(\lambda_k)\,\hat{q}(\lambda_k)\,\hat{\underline{\phi}}_3(\lambda_k)\,\underline{\tilde{\omega}}(k)\right\|^2. \tag{15}$$

The solution to this problem is as follows.

Lemma 2 *A minimizing Q-parameter for problem (15) is*

$$\hat{q}(\lambda)=\sum_{n=0}^{M-1} r(n)\,\lambda^n \tag{16}$$

where

$$r(n)=\frac{1}{M}\sum_{k=0}^{M-1}\tilde{r}(k)\,\lambda_k^{-n}, \tag{17}$$

[that is, $r(n)\xleftarrow{DFT^{-1}}\tilde{r}(k)$*], and* $\tilde{r}(k)$ *solves*

$$\min_{\tilde{r}(k)} \left\|\hat{\underline{\phi}}_1(\lambda_k)\,\underline{\tilde{\omega}}(k)+\hat{\underline{\phi}}_2(\lambda_k)\,\tilde{r}(k)\,\hat{\underline{\phi}}_3(\lambda_k)\underline{\tilde{\omega}}(k)\right\|, \quad k=0,1,\ldots,M-1. \tag{18}$$

So, one only has to find the solution to M minimization problems in (18). These minimization problems share the general form

$$\min_X \|a+BXc\|, \tag{19}$$

where a and c are complex vectors and B and X are complex matrices. The solution to this problem can be obtained in two steps. First find a solution to

$$\min_x \|a+Bx\|.$$

Then find X by obtaining any solution to $Xc=x$. These two problems are standard least square minimization problems and can be easily coded in MATLAB.

4 Design Example

In this section we present a simple but illustrative example to demonstrate the design technique developed in this work. Consider the feedback setup of Figure 6. The plant P is a second order system with transfer function

$$\hat{p}(s) = \frac{0.3s+1}{(0.25s+1)(s+1)}.$$

The reference signal w is the periodic signal

$$w(t) = 0.6\sin(\frac{2\pi}{10}t) + 0.3\sin(2\frac{2\pi}{10}t) + 0.3\sin(3\frac{2\pi}{10}t)$$

of period $T = 10$. The sampling period is chosen to be $h = 0.5$. Therefore, $M = 20$. We design the controller for zero, one, and nine intersample points, that is, for $N = 1, 2$, and 10. The case $N = 1$ represents the conventional discrete-time design, where the emphasis is put on the error signal only at the sampling instants, whereas for $N = 2$ and $N = 10$, the power of the error signal is computed and minimized for one and nine extra points in between the sampling instants, respectively. Thus one expects to see better tracking in the latter cases.

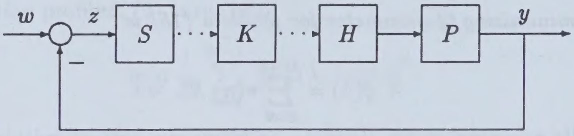


Figure 6: The setup for the design example.

To do the design, we first put the setup of Figure 6 in the form of the setup in Figure 1 and incorporate the technique developed in this work. The steady-state tracking error is shown in Figure 7 for $N = 1, 2$, and 10. As can be seen, the steady-state tracking error is zero at the sampling instants for $N = 1$, but not for $N = 2$ or 10. However, we observe that by taking just one intersample point, i.e., $N = 2$, the overall error reduces dramatically. The error is further decreased if we take an even larger number of intersample points, i.e., $N = 10$. As we take more intersample points, not much improvement in the performance is observed and the error signals almost coincide with that for $N = 10$. This convergence is not surprising as by taking more intersample points, the system in Figure 3 is expected to give a better emulation of the sampled-data setup in Figure 1. We have also listed the power-norm of the steady-state tracking error for different values of N in Table 1 for a quantitative comparison. This table shows that the power of the error reduces almost 50% by taking one intersample point.

Finally, in Figure 8 we have plotted the reference signal w and the output y of the system with the controller designed for $N = 10$ to show the transient response of the system.

N	$\|z_{ss}\|$
1	0.0256
2	0.0134
10	0.0118

Table 1: Power-norm of the steady-state error for different values of N.

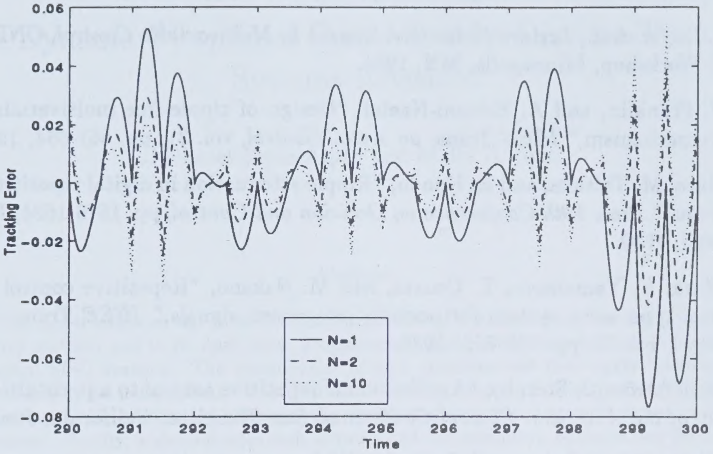


Figure 7: Steady-state tracking error for different number of intersample points.

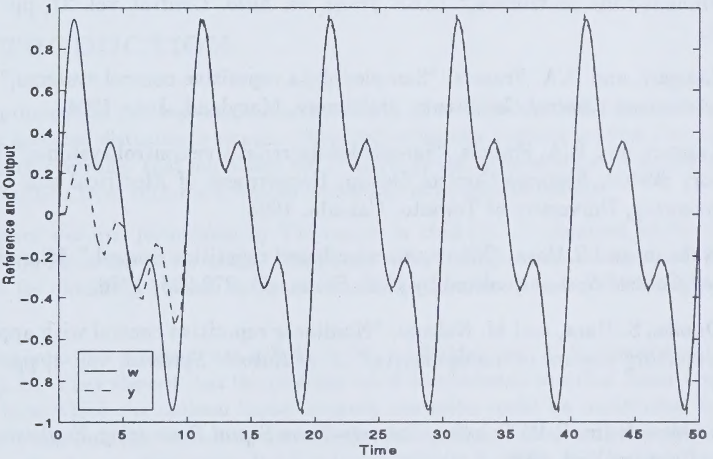


Figure 8: Reference signal and the tracking error for $N = 10$.

References

[1] T. Chen, and B.A. Francis, "Input-output stability of sampled-data systems," *IEEE Trans. on Auto. Control,* vol. 36, pp. 50-59, 1991.

[2] T. Chen, and B.A. Francis, *Optimal Sampled-Data Control Systems,* Springer-Verlag, London, 1995.

[3] K.K. Chew, and M. Tomizuka, "Digital control of repetitive errors in disk drive systems," *IEEE Control System Magazine,* vol. 10, pp. 16-20, 1990.

[4] J.C. Doyle et al., *Lecture Notes in Advances in Multivariable Control,* ONR/Honeywell Workshop, Mineapolis, MN, 1984.

[5] G.F. Franklin, and A. Emami-Naeini, "Design of ripple-free multivariable robust servomechanism," *IEEE Trans. on Auto. Control,* vol. 31, pp. 661-664, 1986.

[6] S. Hara, M. Tetsuka, and R. Kondo, "Ripple attenuation in digital repetitive control systems," *Proc. 29th Conference on Decision and Control,* pp. 1679-1684, Honolulu, Hawaii, 1990.

[7] S. Hara, Y. Yamamoto, T. Omata, and M. Nakano, "Repetitive control systems: A new type servo system for periodic exogenous signals," *IEEE Trans. on Auto. Control,* vol. 33, pp. 659-668, 1988.

[8] G. Hillerstrõm, J. Sternby, "Application of repetitive control to a peristaltic pump," *Proc. of the American Control Conference,* San Francisco, California, June 1993.

[9] T. Inoue, M. Nakano, and S. Iwai, "High accuracy control of servomechanisms for repeated contouring," *Proc. 10th Annual Symp. Incremental Motion Control Systems and Devices,* pp. 258-292, 1981.

[10] J.P. Keller, and B.D.O. Anderson, "A new approach to the discretization of continuous-time controllers," *IEEE Trans. on Auto. Control,* vol. 37, pp. 214-223, 1992.

[11] A. Langari, and B.A. Francis, "Sampled-data repetitive control systems," *Proc. of the American Control Conference,* Baltimore, Maryland, June 1994.

[12] A. Langari, and B.A. Francis, "Sampled-data repetitive control systems," Technical Report #9403, Systems Control Group, Department of Electrical and Computer Engineering, University of Toronto, Canada, 1994.

[13] M. Nakano, and S. Hara, "Microprocessor-based repetitive control," *Microprocessor-Based Control Systems,* edited by N.K. Sinha, pp. 279-296, 1986.

[14] T. Omata, S. Hara, and M. Nakano, "Nonlinear repetitive control with applications to trajectory control of manipulators," *J. of Robotic Systems,* vol. 4, pp. 631-652, 1987.

[15] A.V. Oppenheim, R.W. Schafer, *Discrete-Time Signal Processing,* Englewood Cliffs, N.J., Prentice-Hall, 1989.

[16] M. Tomizuka, T.C. Tsao, and K.K. Chew, "Analysis and synthesis of discrete-time repetitive controllers," *ASME Trans., J. of Dynamic Systems, Measurements, and Control,* pp. 353-358, 1989.

[17] P.P. Vaidyanathan, *Multirate Systems and Filter Banks,* Englewood Cliffs, N.J., Prentice-Hall, 1993.

[18] Y. Yamamoto, "Learning control and related problems in infinite-dimensional systems", *Proc. of the European Control Conference,* pp. 191-222, 1993.

Linear Dynamic ℓ^1-optimal Controllers for Discrete-Time SISO Systems Revisited*

Mehmet Emin Alpay and Molly H. Shor

Abstract

This paper explores the cases when multiplicities of solutions exist for the ℓ^1-optimal control problem and to its dual linear programming problem for discrete-time linear time-invariant SISO systems. The degeneracies of both problems are thoroughly investigated and shown to be independent. It is then shown that a simple solution for the ℓ^1-optimal controller can be constructed whenever the dual linear programming problem has multiple solutions. Finally, a general approach is presented for secondary optimization problems imposed on the set of ℓ^1-optimal controllers for the case where both primary and dual problems have multiple solutions. An example is given of finding the $\mathcal{H}_2$-optimal controller within the set of ℓ^1-optimal controllers.

Keywords - ℓ^1-optimal controller, linear programming, degeneracy, simple solution, secondary optimization problem.

1 INTRODUCTION

The ℓ^1-optimization problem addresses the ability of a linear discrete-time system to reject persistent bounded disturbance signals. The solution for this problem involves the construction of a feedback controller that minimizes the ℓ^1-norm of the impulse response of the closed-loop transfer function from such disturbances to the regulated system output.

This problem was first formulated by Vidyasagar in 1986 [1]. He obtained solutions for SISO discrete time minimum-phase plants and plants with a single unstable zero and proposed a procedure for obtaining bounds on the attainable performances and constructing suboptimal solutions.

The ℓ^1-optimization problem was solved by M. A. Dahleh and J. B. Pearson using duality theory [2], [3]. They showed that the problem could be converted to a dual linear programming problem from which the optimal linear dynamic controller could be constructed by invoking alignment conditions [2]. One of the most important results of this study was the observation that the optimal controller always has a rational transfer function and hence is easily implementable. These results were extended to MIMO systems by the same researchers [3], who then investigated the continuous time equivalent of the problem, the $\mathcal{L}^1$-optimization problem [4].

M. Vidyasagar extended the results of [3] and [4] to the cases where the plant has zeros or poles on the stability boundary (the extended $j\omega$-axis in continuous time, the unit circle in discrete time) and showed that the optimal controller might not exist in such cases [5]. He also proposed an iterative algorithm of constructing suboptimal controllers with performances converging to the unattainable optimal result.

---

*This work was supported in part by the National Science Foundation under Grant ECS-9309336.

D. G. Meyer pointed out that the ℓ^1-optimal controller can be of arbitrarily high order and presented examples that show that the dual linear programming problem need not have a unique solution [6].

M. Dahleh and M. A. Dahleh then discussed this nonuniqueness briefly in [7], pointing out that the scalar (SISO) problem generically has a unique solution and then making use of the nonuniqueness to impose a secondary ℓ^2 criterion on the set of ℓ^1-optimal solutions, when the linear programming problem does have multiple solutions. This produces a quadratic programming problem, which always has a unique solution. They then discuss the continuity of the resulting "super optimal" controllers in order to propose an adaptive control scheme based on ℓ^1-optimal controllers.

Our contribution is a more complete, more careful, study of the degeneracy properties of both the primal ℓ^1-optimal control problem and of the dual linear programming problem. We approach this study using basic Euclidean geometric arguments, rederiving some of the original results and clarifying the consequences for degeneracy of the two problems and for secondary optimization problems imposed on the set of ℓ^1-optimal controllers. The outlines of direct proofs are given for the existence of solutions for both the primary and dual problems (see [8] for details). Conditions for the existence of multiple solutions for each of these problems are then thoroughly investigated using geometric arguments, and consequences of the existence of multiple solutions are explored.

The following new, perhaps surprising, results are obtained. Either the primal or the dual problem may have multiple solutions while the other has a unique solution. These degeneracies are, in fact, unrelated. A simple solution can always be constructed for the ℓ^1-optimal controller whenever the dual linear programming problem has multiple solutions. The solution for the ℓ^1-optimal controller can always be found using a corner solution of the linear programming problem, even when the linear programming problem admits solutions on an entire face of the convex constraint set.

For the case when multiple solutions exist for the primal problem, a secondary performance criterion can be imposed on the set of ℓ^1-optimal controllers (as proposed in [7]). The solution of such a secondary optimization problem can always be found using a corner solution of the dual problem even when it has multiple solutions. This provides a computational limit on the number of secondary problems to be solved. A general approach is given for the solution of such problems and is illustrated for the $\mathcal{H}_2$-optimality criterion, resulting in a quadratic programming problem.

2 MATHEMATICAL PRELIMINARIES

In this section, we introduce the mathematical terms and give the statements of the technical lemmas and theorems that are used throughout the rest of this paper.

Definition A causal sequence $\{x(k)\}$ is called an *ℓ^1-sequence* if it is absolutely summable; i.e., $||x||_1 = \sum_{k=0}^{\infty} |x(k)| < \infty$, where $||x||_1$ denotes the *ℓ^1-norm* of the sequence. The normed linear space of ℓ^1-sequences is called the *ℓ^1 space.*

Definition A causal sequence $\{x(k)\}$ is called an *ℓ^∞-sequence* if it is bounded; i.e., $||x||_\infty = \sup_{k\geq 0} |x(k)| < \infty$, where $||x||_\infty$ denotes the *ℓ^∞-norm* of the sequence. The normed linear space of ℓ^∞-sequences is called the *ℓ^∞ space.*

Lemma 1 *Let $x \in \ell^\infty$, $h \in \ell^1$ and $y = x * h$. Then (1) $y \in \ell^\infty$, (2) $||y||_\infty \leq ||h||_1 ||x||_\infty$, and (3) $\sup_{||x||_\infty \leq 1} \frac{||y||_\infty}{||x||_\infty} = ||h||_1$.*

Definition Let X be a normed linear space. Its *dual space* is defined as the space of linear continuous functionals on X and is denoted by X^*. The *norm of a functional* $x^* \in X^*$ is defined as $\sup_{||x|| \leq 1} |x^*(x)| = \sup_{||x|| \leq 1} | < x, x^* > |$, where $< x, x^* >$ denotes $x^*(x)$.

Lemma 2 *Any continuous linear functional f on ℓ^1 can be represented by an ℓ^∞-sequence y such that $f(x) = \sum_{k=0}^{\infty} x(k)y(k)$ for all $x \in \ell^1$. In this sense, ℓ^∞ is the dual space of ℓ^1.*

Lemma 3 *Let $x \in \ell^1$ and $r \in \ell^\infty$. Then x and r are aligned if and only if (1) $x(k)r(k) \geq 0$ for all $k = 0, 1, \ldots$ and (2) $x(k) = 0$ whenever $|r(k)| \neq ||r||_\infty$.*

Theorem 1 *(Duality Theorem) Let x be an element in a real normed linear space X, and let μ denote its distance from a subspace S of X. Then $\mu = \inf_{k \in S} ||x - k|| = \max_{r \in BS^\perp} < x, r >$, where $BS^\perp = \{x^* \in X^* | < x, x^* >= 0$ for all $x \in S$ and $||x^*|| \leq 1 \}$. The maximum on the right is achieved for some $r_0 \in BS^\perp$ with $||r_0|| = 1$. Moreover, if the infimum on the left is achieved for some $x_0 \in S$, then r_0 is aligned with $x - x_0$.*

Definition Let $b_i \in \Re^n$, $i = 1, \ldots, m$ be n-dimensional real vectors. A *sub-convex combination* of these vectors is defined as a linear combination of the form $\beta_1 b_1 + \beta_2 b_2 + \cdots + \beta_m b_m$ where β_i satisfy the condition $\sum_{i=1}^{m} |\beta_i| \leq 1$.

Lemma 4 *Let $x \in \Re^n$ be such that $|b_i' x| \leq 1$ for $i = 1, \ldots, m$ with $b_i \in \Re^n$. An additional inequality $|b_{m+1}' x| \leq 1$ is automatically satisfied if and only if b_{m+1} can be expressed as a sub-convex combination of the b_i, $i = 1, \ldots, m$. If this is the case, $|b_{m+1}' x| \leq 1$ will be called an* **irrelevant inequality**.

Lemma 5 *Suppose there are $N \geq n$ inequalities of the form $|b_i' x| \leq 1$ where $b_i, x \in \Re^n$ and $\{b_i\}_{i=1}^n$ forms a basis for $\Re^n$. Then the number of relevant linearly independent inequalities is n.*

3 Review of ℓ^1-optimal control problem and its dual linear programming problem

We begin by reviewing the formulation of the ℓ^1-optimal control problem and the approach given by M. A. Dahleh and J. B. Pearson to convert this problem to an equivalent linear programming problem.

The system and feedback equations for the linear time-invariant SISO discrete-time system considered in this paper are given below

$$\begin{aligned} Z &= G_{11}W + G_{12}U \\ Y &= G_{21}W + G_{22}U \\ U &= KY, \end{aligned} \tag{1}$$

Here, w is the disturbance, u is the input signal, y is the measured output, and z is the regulated output. Capital letters denote the unilateral z-transforms of these signals, defined by $X(z) = X = \sum_{k=0}^{\infty} x(k)z^k$. where the G_{ij} are real rational stable transfer functions.

The ℓ^1-optimal controller is the controller that minimizes the ℓ^∞-induced norm $\sup_{||w||_\infty \leq 1} \frac{||z||_\infty}{||w||_\infty}$, and hence minimizes the effect of the disturbance on the output, keeping the closed-loop system stable using stable controllers. From Lemma 1, this induced norm is identical to the ℓ^1-norm of the impulse response corresponding to the transfer function from the disturbance to the output. Thus, the ℓ^1-optimal control problem is to find a stabilizing dynamic feedback that minimizes $||\phi||_1$, where

$$\phi = Z^{-1}\{\Phi\} = Z^{-1}\left\{G_{11} + \frac{KG_{21}G_{12}}{1 - G_{22}K}\right\}.$$

By using the YJBK parameterization of all stabilizing controllers [9], [10], it can be shown [3], [4] that Φ can always be expressed as $\Phi = H - FQ$, where Q is any stable transfer function. The transfer functions H and F are obtained from the system transfer functions G_{ij}, $i, j = 1, 2$, as $H = G_{11} + G_{12}G_{21}YM$ and $F = G_{12}G_{21}M^2$, where $G_{22} = N/M$ such that M and N are stable, coprime, and satisfy Bezout's identity $MX - NY = 1$ for some stable X and Y.

The problem now is reduced to that of finding an appropriate stable transfer function Q that minimizes $||\phi||_1$. The controller block K can then be constructed as $K = (X - MQ)/(Y - NQ)$.

Define $R = FQ$. The only restriction on the transfer function Q is that it be stable, which is equivalent to its having all its poles outside the unit circle. Thus, any zeros of F inside the unit circle, a_i, $i = 1, \ldots, n$, cannot be canceled by the poles of Q. For convenience, assume that all of these zeros have single multiplicities. Then R can be any stable transfer function with $R(a_i) = 0$ for all $i = 1, \ldots, n$.

The condition of stability on R is equivalent to $r = Z^{-1}\{R\}$ being an ℓ^1-sequence. Hence, if the second set of constraints on R, i.e., $R(a_i) = 0$, $i = 1, \ldots, n$, is converted to a constraint set in the time-domain, then the whole problem can be solved as an ordinary minimum norm problem in the ℓ^1 space, as was done by M. A. Dahleh and J. B. Pearson [3], [4]. These constraints define a subspace in ℓ^1, and the problem is reduced to finding the minimum distance from h to this subspace and finding the ℓ^1 sequence r for which it is achieved - if it exists. This is the dual linear programming problem, which can be stated as in the following theorem:

Theorem 2 *(M. A. Dahleh and J. B. Pearson [2]) Let a_i, $i = 1, \ldots, n$ be the zeros of F inside the unit circle, ordered such that $a_i \in \Re$, $i = 1, \ldots, m$ and $a_i = \bar{a}_{i+l}$, $i = m+1, \ldots, m+l$, with $m + 2l = n$; $R = FQ$, where Q is a stable, real, rational transfer function; and F be obtained from the YJBK parameterization of the transfer function Φ. Then the solution of the following linear programming problem results in the ℓ^1-optimal performance μ of the system (1):*

$$\begin{aligned} \mu &= \inf_{R \in S} ||h - r||_1 \\ &= \max_{\alpha_i} \left\{ \sum_{i=1}^{m} \alpha_i H(a_i) + \sum_{i=m+1}^{m+l} \alpha_i \Re(H(a_i)) + \sum_{i=m+l+1}^{n} \alpha_i \Im(H(a_{i-l})) \right\} \end{aligned} \tag{2}$$

subject to the constraints

$$-1 \leq \sum_{i=1}^{m} \alpha_i a_i^k + \sum_{i=m+1}^{m+l} \alpha_i \Re(a_i^k) + \sum_{i=m+l}^{n} \alpha_i \Im(a_{i-l}^k) \leq 1$$

for all $k = 0, 1, 2, \ldots$. S denotes the set of stable transfer functions that satisfy $R(a_i) = 0$ for $i = 1, \ldots, n$.

4 SOLUTION OF THE ℓ^1-OPTIMIZATION PROBLEM

In this section, we give a brief overview of the solution of the ℓ^1-optimization problem, using geometric arguments. We first study the dual linear programming problem and prove that it is always solvable via available numerical techniques and always leads to a finite result. Then we study the problem of constructing the ℓ^1-optimal controller from the solution of the dual linear programming problem and show by construction that an ℓ^1-optimal controller always exists.

The approach presented herein results in geometrical insight into the problem in the Euclidean $\Re^n$ space, which is useful later in the study of the degeneracy of the two problems.

4.1 Solution of the Linear Programming Problem

To show that the dual linear programming problem, from Theorem 2 and [2], can always be solved using available numerical techniques and leads to a finite result, we

- show that only finitely many constraint inequalities are relevant, since infinitely many constraints cannot be handled by computer algorithms,
- find a criterion to distinguish relevant constraints from the irrelevant ones,
- show that the constraint region denoted by the relevant inequalities is a compact one, so that a finite solution for the linear programming problem always exists.

The first condition was stated in [2] and proved for the MIMO case in [3]. The second and third conditions are direct consequences of Lemma 4 and 5.

Lemma 4 provides a way of distinguishing between relevant and irrelevant inequalities as follows. The first n inequalities are linearly independent and relevant within themselves. Any additional inequality should be eliminated if it can be expressed as a sub-convex combination of the previous ones. It must also be checked if a new inequality eliminates some of the previous ones. From Lemma 5, if we start with n linearly independent inequalities, then we will end up with n linearly independent relevant ones even after all the eliminations. Combined with the fact that each inequality constraint defines a region between two parallel hyperplanes symmetric about the origin, it follows that the constraint region denoted by the relevant constraints is a compact one in $\Re^n$, which guarantees the existence of a finite solution to the linear programming problem.

The irrelevant inequalities do not affect the constraint region and hence have no influence on the solution of the linear programming problem. However, in some pathological cases, the boundary of the region denoted by an irrelevant inequality intersects the constraint region, an example of which was provided by D. G. Meyer [6]. If the solution of the linear programming problem happens to lie within this intersection, an extra degree of freedom in the construction of the ℓ^1-optimal controller is introduced. As a consequence, an inequality might affect the ℓ^1-optimal controller even when it is an irrelevant one.

4.2 Construction of an ℓ^1-optimal Controller

In the previous section, we verified that the linear programming problem associated with the ℓ^1-optimization problem always has a finite solution that can be computed using available numerical techniques. The existence of solutions was known from Dahleh and Pearson's work.

This finding does not immediately tell if a solution to the original problem of constructing an ℓ^1-optimal controller exists and, if it does, how it can be constructed.

In this section, we give a brief overview of the conditions that the closed-loop transfer function Φ has to satisfy to give the ℓ^1-optimal performance and prove that a controller yielding this result can always be constructed.

This new constructive proof will give us the geometric interpretation of the linear programming problem and construction of ℓ^1-optimal controllers that will be used to study the degeneracy of the two problems.

Let $[\alpha_1^*, \ldots, \alpha_n^*]'$ be a solution for the linear programming problem. Define

$$w_0(k) = \sum_{i=1}^{m} \alpha_i^* a_i^k + \sum_{i=m+1}^{m+l} \alpha_i^* \Re(a_i^k) + \sum_{i=m+l+1}^{n} \alpha_i^* \Im(a_{i-l}^k)$$

Now assume that the infimum on the left in Theorem 2 is actually achieved for some r_0. Then, defining $\phi_0 = h - r_0$, it can be shown (Theorem 1 and Lemma 3) that the necessary and sufficient conditions for the existence of an optimal solution are

1. $\phi_0(k) w_0(k) \geq 0$
2. $\phi_0(k) = 0$ whenever $|w_0(k)| \neq \|w\|_\infty = 1$
3. $\sum_{k=0}^{\infty} |\phi_0(k)| = \mu$
4. $\Phi_0(a_i) = \sum_{k=0}^{\infty} \phi_0(k) a_i^k = H(a_i)$ for $i = 1, \ldots, n$

The first two conditions are alignment requirements. The third one assures ϕ achieves the infimum norm, and the last condition arises from the hypothesis $\Phi(a_i) = H(a_i) - R(a_i) = H(a_i)$.

Condition 2 restricts the indices for which $\phi_0(k)$ can be nonzero. Recalling the definition of w_0, it is evident that $\phi_0(k)$ can be nonzero only on the parallel constraint surfaces denoted by the equations

$$\sum_{i=1}^{m} \alpha_i a_i^k + \sum_{i=m+1}^{m+l} \alpha_i \Re(a_i^k) + \sum_{i=m+l+1}^{n} \alpha_i \Im(a_{i-l}^k) = \pm 1$$

for those indices k corresponding to active constraints. Since the corner $[\alpha_1^*, \ldots, \alpha_n^*]'$ is specified by the intersection of $q \geq n$ such surfaces obtained from different values of k, we will have q possibly nonzero entries of ϕ_0 which should be adjusted to satisfy the remaining conditions.

From the next theorem, it follows a solution ϕ_0 to this problem satisfying all the conditions always exists, which immediately implies that an ℓ^1-optimal linear dynamic controller can always be constructed.

Theorem 3 *At any corner $[\alpha_1^*, \ldots, \alpha_n^*]'$ that maximizes the cost functional of the linear programming problem associated with the ℓ^1-optimization problem, a solution ϕ_0 satisfying existence conditions for an optimal solution, Conditions 1-4 above, can always be constructed.*

Corollary 3.1 *For a discrete-time SISO system, an ℓ^1-optimal linear dynamic controller always exists.*

Proof An outline of the proof of Theorem 3 is as follows. First, we show that any solution that satisfies Conditions 1, 2 and 4 automatically satisfies Condition 3 as well. Next, we show that a solution satisfying Conditions 2 and 4 can always be found. Finally, we show that if a solution satisfies Conditions 2 and 4 and is unique, then it automatically satisfies Condition 1. If there are multiple solutions for 2 and 4, we show that we can still construct one satisfying Condition 1 as well. The proof of the corollary follows immediately by recalling that $\phi_0 = h - r_0 \Rightarrow r_0 = h - \phi_0$, from which the construction of the ℓ^1-optimal controller is straightforward.
Q.E.D.

5 DEGENERATE CASES

In this section, we study the relationship between the degeneracies of the ℓ^1-optimization problem and the dual linear programming problem and investigate the consequences of there being multiple solutions to the dual linear programming problem. We obtain necessary and sufficient conditions for the existence of multiple solutions to the dual linear programming problem associated with the ℓ^1-optimization problem and prove that a simple solution to the original problem can be constructed in such cases. We show that the ℓ^1-optimal controller can always be found using a corner solution to the linear programming problem. We discuss necessary conditions for the existence of multiple solutions to the primary problem. Except for some rare cases, the ℓ^1-optimal linear dynamic controller is unique, even when the dual linear programming problem has multiple solutions. We show that the degeneracies of the primary and dual problems are completely independent from each other. We also prove an important theorem regarding how to approach secondary optimization problems imposed on ℓ^1-optimal controllers in the off-chance that both the primary and dual problem have multiple solutions. The solution of such a secondary optimization problem can always be found using a corner solution to the linear programming problem.

5.1 Multiple Solutions for the Linear Programming Problem

Define

$$\Psi = [H(a_1), \ldots, H(a_m), \Re(H(a_{m+1})), \ldots, \Re(H(a_{m+l})), \Im(H(a_{m+1})), \ldots, \Im(H(a_{m+l}))]'$$

If α^* denotes a corner that is the intersection of $q \geq n$ constraint hyperplanes of the form

$$\sum_{i=1}^{m} \alpha_i a_i^{k_j} + \sum_{i=m+1}^{m+l} \alpha_i \Re(a_i^{k_j}) + \sum_{i=m+l+1}^{n} \alpha_i \Im(a_{i-l}^{k_j}) = s_j$$

for $j = 1, \ldots, q$ and is a solution for the linear programming problem associated with the ℓ^1-optimization problem, a necessary and sufficient condition for the existence of multiple solutions to the linear programming problem is Ψ being a linear combination of $p < n$ of the normals to these hyperplanes; i.e., defining $b_j = [a_1^{k_j} \ldots a_m^{k_j} \, \Re(a_{m+1}^{k_j}) \ldots \Re(a_{m+l}^{k_j}) \, \Im(a_{m+1}^{k_j}) \ldots \Im(a_{m+l}^{k_j})]'$, we should have $\Psi = c_1 b_{j_1} + \cdots + c_p b_{j_p}$. Then

$$\Psi'\alpha^* = c_1 b'_{j_1}\alpha^* + \cdots + c_p b'_{j_p}\alpha^* = \sum_{i=1}^{p} c_{j_i} s_{j_i}$$

from above. Furthermore, since the cost functional $\alpha'\Psi$ achieves its maximum at α^*, the c_{j_i} have the same signs as s_{j_i}, yielding the ℓ^1-optimal performance index $\mu = \sum_{i=1}^{p} |c_{j_i}|$.

In such a case, there is a simple solution to the ℓ^1-optimization problem, as stated below.

Theorem 4 *In the presence of multiple solutions to the linear programming problem, there is a simple solution to the ℓ^1-optimization problem.*

Proof The proof is constructive and merely consists of showing the simple closed-loop impulse response

$$\phi(k) = \begin{cases} c_j & k = k_j \\ 0 & \text{otherwise} \end{cases}$$

satisfies all the necessary and sufficient conditions (Conditions 1–4 in Section 4.2) for qualifying as an optimal solution. The ℓ^1-optimal controller is then obtained in a straightforward manner by setting $r = h - \phi$. Q.E.D.

5.2 Multiple Solutions for the ℓ^1-optimal Controller

From Theorem 4, if the solution of the linear programming problem is at a corner of the constraint set at which exactly n active constraint surfaces intersect, then there is a unique solution for the ℓ^1-optimization problem. From Theorem 5, if there are multiple solutions to the linear programming problem, then a simple solution to the ℓ^1-optimization problem can be constructed. Recalling that an ℓ^1-optimal closed-loop impulse response ϕ must satisfy Conditions 1–4 in section 4.2, we see that the entries of ϕ must satisfy n linear equations and ϕ has q possibly non-zero entries with sign restrictions on them to satisfy these equations, where q is the number of constraint surfaces passing through the point α^* that maximizes the linear programming cost functional. Clearly, a necessary but not sufficient (due to sign restrictions) requirement for the existence of multiple solutions to the ℓ^1-optimization problem is that the system of n linear equations mentioned above admits multiple solutions; i.e., the matrix equation

$$AP = \Psi$$

must have multiple solutions, where P is a $q \times 1$ vector of the possibly nonzero components of the impulse response ϕ, and A and Ψ can be directly obtained from Condition 4 in section 4.2. There are two possibilities:

Case 1 If the linear programming problem has a unique solution, α^* is a corner and $q \geq n$, with A matrix having full row-rank. Thence, the above system of equations is always consistent and the necessary and sufficient condition for the existence of multiple solutions is $q > n$.

Case 2 If the linear programming problem is degenerate, α^* can be any point in the set of parameters solving that problem. Nevertheless, for any non-corner α^*, there exists a corner through which all the surfaces that pass through α^* pass, as well as at least one more. Hence, the necessary and sufficient condition for the existence of multiple solutions in this case is $q > n$ at any corner of the solution set to the linear programming problem.

Thus, a necessary but not sufficient condition for the existence of multiple solutions to the ℓ^1-optimization problem is that $q > n$ surface hyperplanes pass through a corner α^* that solves the linear programming problem. This is equivalent to the $q \times n + 1$ matrix constructed from these q hyperplane equations in $\Re^n$ having rank n (i.e. not full-rank), which is highly unlikely to happen since even the slightest disturbance of one of the entries might disrupt it. Consequently, except for some rare cases, the ℓ^1-optimal controller will be unique.

A careful interpretation of the arguments in this and the previous section reveals the interesting fact that the degeneracy of the ℓ^1-optimization problem is completely independent from that of the linear programming problem associated with it, where the term *degeneracy* refers to the existence of multiple solutions for the respective problem. In the off-chance that both problems have multiple solutions, a secondary optimization problem imposed on the set of ℓ^1-optimal controllers can be treated as discussed below.

Theorem 5 *Assume both the primary and dual problems are degenerate. Then the linear programming problem parameters (α_i) corresponding to an ℓ^1-optimal controller optimizing any secondary performance criterion imposed on the closed-loop transfer function Φ can be found at a corner of the solution set for the linear programming problem.*

Proof The proof is merely argumentative. Recall that that the closed-loop impulse response ϕ has to satisfy Conditions 1–4 in Section 4.2 for optimal performance. Condition 2 tells us that $\phi(k)$ can be nonzero only if the linear programming parameters α_i lie on the constraint surface denoted by

$$\sum_{i=1}^{m} \alpha_i a_i^k + \sum_{i=m+1}^{m+l} \alpha_i \Re(a_i^k) + \sum_{i=m+l}^{n} \alpha_i \Im(a_{i-l}^k) = \pm 1$$

Assume that both the primary and dual problems are degenerate and a secondary optimization problem is imposed on the ℓ^1-optimal controllers. Our aim is to pick the α_i in the solution set for the linear programming problem that would lead to the solution of this secondary problem. However, for each non-corner point $[\alpha_1, \dots, \alpha_n]'$ in the solution set, there is a corner point of that set where all the equations for $[\alpha_1, \dots, \alpha_n]'$ are satisfied, plus at least one additional equation. Since both vectors would lead to identical sign constraints on the entries of ϕ, the result is an effective loss of degree of freedom in choosing the entries of ϕ. Thus, the linear programming parameters corresponding to an ℓ^1-optimal controller optimizing a secondary performance criterion can always be found at a corner of the solution set for the linear programming problem. Q.E.D.

An example follows. When there are multiple solutions to the ℓ^1-optimal controllers, a $\mathcal{H}_2$-optimization problem imposed on the set of ℓ^1-optimal controllers can always be expressed as:

$$\text{minimize } ||\phi||_2 = \left(\sum_{k=0}^{\infty} |\phi(k)|^2 \right)^{1/2} = P'P \text{ subject to the constraints } AP = H \text{ and } P_i s_i \geq 0$$

where A, P and H are as defined in the previous sections.

The problem as formulated above can easily be solved by applying any of the quadratic programming algorithms available in the literature if the matrix A is well-defined. In cases where both the primary and dual problems have multiple solutions, however, the entries of this matrix will change with respect to the α point picked up as the solution of the dual problem. The main contribution of Theorem 5 becomes evident at this point: without this result we would have to try many different settings for various choices of the α vectors in the solution set that yield the ℓ^1-optimal performance. But Theorem 5 enables us to concentrate on the solutions at the corners alone and greatly simplifies the computation process.

6 CONCLUSION

In this paper, we have thoroughly examined the cases when the ℓ^1-optimal control problem or its dual linear programming problem have multiple solutions, using basic Euclidean geometric arguments. We also gave direct proofs of the existence of solutions to both problems, using basic arguments. We discussed how a secondary optimization problem could be imposed on the class of ℓ^1-optimal controllers, and showed that the solution of such a problem can always be found using a corner solution of the linear programming problem.

References

[1] M. Vidyasagar, "Optimal rejection of persistent bounded disturbances," *IEEE Transactions on Automatic Control*, vol. AC-31, June 1986.

[2] M. A. Dahleh and J. B. Pearson, Jr., "ℓ^1 optimal controllers for discrete time systems," in *Proceedings of the ACC*, (Seattle, WA), June 1986.

[3] M. A. Dahleh and J. B. Pearson, Jr., "ℓ^1 optimal feedback controllers for mimo discrete-time systems," *IEEE Transactions on Automatic Control*, vol. AC-32, April 1987.

[4] M. A. Dahleh and J. B. Pearson, Jr., "$\updownarrow^1$ optimal compensators for continuous-time systems," *IEEE Transactions on Automatic Control*, vol. AC-32, October 1987.

[5] M. Vidyasagar, "Further results on the optimal rejection of persistent bounded disturbances," *IEEE Transactions on Automatic Control*, vol. 36, June 1991.

[6] D. G. Meyer, "Two properties of ℓ^1-optimal controllers," *IEEE Transactions on Automatic Control*, vol. AC-33, September 1988.

[7] M. Dahleh and M. A. Dahleh, "Optimal rejection of persistent bounded disturbances: Continuity p roperties and adaptation," *IEEE Transaction on Automatic Control*, vol. AC-35, June 1990.

[8] M. E. Alpay, "Existence and uniqueness results for ℓ^1-optimal linear dynamic controllers for discrete time siso systems," Tech. Rep. 95.005, Oregon State University, August 1995.

[9] M. A. Dahleh and I. J. Diaz-Bobillo, *Control of Uncertain Systems: A Linear Programming Approach.* Englewood Cliffs, NJ: Prentice Hall, 1995.

[10] D. G. Luenberger, *Optimization by Vector Space Methods.* New York, NY: J. Wiley & Sons, 1969.

Utilization of the MV Spectrum for Diagnosis and Estimation of Harmonic Signals

Andrew Sexton. and Donald E. Lyon
Dept. Mech. Eng., Univ. of New Brunswick
Fredericton, New Brunswick E3B 5A3

Abstract:

This paper considers the identification of a single channel two-dimensional harmonic signal field contained in an unknown and possibly nonstationary noise field. A technique is presented using the convergence based point spectra developed by Frazho and Sherman [1]. This technique has the advantage of discerning between point and narrowband noise spectrum based on the monotonic convergence properties of a sequence of Minimum Variance (MV) spectra. A comprehensive example is presented demonstrating the properties and utility of the approach along with a comparison to the multi-dimensional Discrete Fourier Transform (m-D DFT) and Autoregressive (AR) modelling based techniques.

1. Introduction:

The problem of identifying harmonic components of a single channel random field arises in many applications, such as radar and sonar processing, image processing, active noise control, and wave analysis to name a few. Compared to the variety of random process techniques, the number of practically applicable random field techniques for spectral estimation is limited. This is because the extension of many random process techniques to the random field is not practical due to computational complexity or the inextensibility of the one-dimensional notion of polynomial roots [2]. The resulting random field techniques which are available can generally be fit into two families: m-D DFT and m-D polynomial techniques [2].

The m-D DFT is the most popular technique for spectral analysis of random fields. The m-D DFT can be considered as either a continuous spectrum estimator, since it can be expressed as a m-D Z-transform approximation, or a point spectrum estimator, since it can be viewed as a Hilbert space projection on a finite basis of sinusoids. However, it does not have validity for both estimators simultaneously. This is because the projection of a sinusoid with a frequency different from each basis vector frequency will result in the sinusoid being expressed as a linear combination of the basis sinusoids. This, of course, is the effect of spectral leakage wherein energy at a particular frequency falsely appears at other frequencies. Assuming data is taken using a rectangular window then the leakage associated with point spectra corresponds to convolving the true point spectrum with an m-D sinc function in the frequency domain. Of course, spectral leakage is also a concern for continuous spectra since any energy at non-basis frequencies must be "leaked" to the basis frequencies. The issue of spectral leakage is very significant in the multi-dimensional setting as it can lead to spectral plots which are very difficult to interpret as will be shown in the later example.

Relative to using the m-D DFT for harmonic signal identification, the most common approach is to associate the largest spectral peaks in the autospectra of the estimated spectral density matrix with sinusoids. Then the power associated with a sinusoid is estimated by local integration under the peak. This may be suitable for white noise corruption where sinusoidal signals are not closely spaced. However, without knowledge of the noise colour, there is no basis for claiming that any given peak is associated with a sinusoid, as opposed to a narrowband noise component. Furthermore, accurate power estimates can be very difficult. Hence, the DFT is not appropriate as a simultaneous continuous and point spectrum estimator.

Multi-dimensional polynomial techniques based on the m-D Yule-Walker equations or the Maximum Entropy Method are also common [2]. These techniques are high resolution, continuous spectrum approaches. Hence, they also suffer from spectral leakage for harmonic signals, which as before can make interpretation of the spectral information difficult. In particular, since the polynomials associated with these techniques are typically zero-padded to improve visualization of the spectra, the zero-padding interpolates the sinc-function associated with leakage and thus can produce a multitude of large false peaks. Furthermore, the false peaks caused by leakage can interact with true peaks to yield spectral plots with misleading information and render efforts to estimate power by local integration useless. A final concern for these techniques relates to their ability to handle large numbers of sinusoids since it is common for polynomial orders to be low relative to the complexity of the field. The number of polynomial orders required to model a sinusoid is two times the dimension of random field. Insufficient polynomial orders to represent each sinusoid can result in spectral components being smeared together, which, in turn, can result in flawed conclusions regarding a field's character.

Recently, Frazho and Sherman [1] proposed a new spectral-based technique for recovering a multichannel harmonic signal field in an arbitrary, unknown, and possibly nonstationary noise field. This technique is an extension to the work of Foias et al. [3] for recovering a harmonic signal embedded in an arbitrary, unknown multichannel random noise process. The techniques of both [1] and [3] derive their power by utilizing the monotonic convergence properties of a sequence of MV spectra. In the multichannel random process setting, this technique has been successfully investigated and found to have notable advantages over traditional approaches [4],[5],[6],[7]. In particular, the work of [4] demonstrated that in the limit the MV eigenspectrum or the MV autospectrum will decrease by 3 dB at noise frequencies for each doubling of polynomial order. This property has formed the basis for a simple convergence test to automatically discern between narrowband noise and harmonic signal peaks. Moreover, since the MV spectrum is a true power spectrum, not a power spectral density, the heights of the convergent signal peaks can yield notably more reliable estimates of the signal power [5]. In this work we will demonstrate that these properties readily extend to the multichannel, multidimensional setting and we will present a single channel two-dimensional image processing example to demonstrate these properties. Moreover, we will show that the multi-dimensional extension of this technique retains, and in fact enhances, its random process advantages over conventional spectral approaches. In particular, we will show that the method does not suffer from any of the issues concerning the m-D DFT or the m-D polynomial based techniques discussed above. Finally, we remark that the spectrum utilized here is in essence a rediscovery of the MV spectrum associated with Capon [2]. However, the spectrum differs in terms of philosophy, in that a sequence of MV spectra are used.

The following section presents the fundamental equations for the MV spectra and the AR spectra, which is a m-D polynomial technique. Section 2 also discusses some properties associated with the MV spectrum. Section 3 briefly presents computational issues regarding the proposed technique. Section 4 then presents a comprehensive example demonstrating the proposed technique and comparing it to standard DFT and AR spectral approaches. Finally, Section 5 presents a brief discussion and summary of our results.

2. Theoretical Background

This section briefly reviews theory associated with the MV and AR spectra. To begin, define $\{\sigma_i\}_{i=1}^{n}$ to be an ordering of the group G, the integer valued coordinates on $\mathbb{R}^d$. Consider a p-channel, d-dimensional random field, $\mathbf{Y}(\sigma)$, of the form,

$$\mathbf{Y}(\sigma) = \sum_{j=1}^{q} \mathbf{A}_j[e^{-i[(\gamma_j,\sigma) + \xi_{1j}]}, \ldots ,e^{-i[(\gamma_j,\sigma) + \xi_{pj}]}] + \mathbf{N}(\sigma) \equiv \mathbf{S}(\sigma) + \mathbf{N}(\sigma)$$

where the random signal field, $\mathbf{S}(\sigma)$, consists of q sinusoids with distinct wavenumbers, γ_j, non-zero amplitude matrices with complex entries, $\mathbf{A}_j$, and uniform iid phases over $[0,2\pi)$, ξ_{ij}. $(\gamma,\sigma) = \sigma^{\mathrm{tr}}\gamma$ denotes the usual inner product on $\mathbb{R}^d$. The noise, $\mathbf{N}(\sigma)$, is assumed to be a homogeneous random noise field with correlation function $\{\mathbf{R}_N(\alpha,\beta)\}_{\alpha,\beta\in G} = \{\mathbf{R}_N(\alpha\text{-}\beta)\}_{\alpha,\beta\in G}$. The correlation sequence for $\mathbf{Y}(\sigma)$ is

$$\mathbf{R}_\mathbf{Y}(\alpha,\beta) = \sum_{j=1}^{i} \mathbf{A}_j\mathbf{A}_j^* c^{-i(\gamma_j,\alpha-\beta)} + \mathbf{R}_\mathbf{N}(\alpha-\beta) \equiv \mathbf{R}_\mathbf{S}(\alpha-\beta) + \mathbf{R}_\mathbf{N}(\alpha-\beta) \equiv \mathbf{R}_\mathbf{Y}(\alpha-\beta) .$$

The $\mathbf{AR}_n(\omega)$ model for $\mathbf{Y}(\sigma)$ has coefficients $\{-\alpha_n(\sigma_j)\}_{j=1}{}^n$ which solve the m-D Yule-Walker system of equations

$$\mathbf{R}_n[1,\alpha_n(\sigma_1),\ldots,\alpha_n(\sigma_n)]^t = [\varepsilon_n\varepsilon_n^*,0,\ldots,0]^t .$$

Here $\mathbf{R}_n = \{\mathbf{R}_\mathbf{Y}(\sigma_i,\sigma_j)\}_{i,j=0}{}^n$ denotes the $(n+1)\times(n+1)$ correlation matrix and t denotes transpose. For any $\beta=(\beta_1,\ldots,\beta_d) \in G$ and $z=[\lambda_1,\ldots,\lambda_d]$ in $\mathbb{C}^d$, denote

$$z^\beta = \lambda_1^{\beta_1}\cdots\lambda_d^{\beta_d}$$

and assume that z is a vector in $\mathbb{C}^d$ whose components have modulus one; that is $|\lambda_i|=1$. Letting

$$z = e^{i\omega} = [e^{i\omega_1},e^{i\omega_2},\ldots,e^{i\omega_d}] , \quad \omega = [\omega_1,\omega_2,\ldots,\omega_d] ,$$

we can now define the nth order normalized m-D polynomial $\Phi(e^{i\omega})$ by

$$\Phi_\mathbf{n}(e^{i\omega}) = \Phi_\mathbf{n}(z) = [\sum_{k=0}^{n} \alpha_n(\sigma_k)e^{i(\omega,\sigma_n-\sigma_k)}]\boldsymbol{\varepsilon}_n^{-1} = [\sum_{k=0}^{n} \alpha_n(\sigma_k)z^{(\sigma_n-\sigma_k)}]\boldsymbol{\varepsilon}_n^{-1}, \quad \alpha_n(\sigma_0) = \mathbf{I} .$$

The $\mathbf{AR}_n(\omega)$ spectral representation for $\mathbf{Y}(\sigma)$ is then

$$\mathbf{AR}_n(\omega) = [\Phi_n(e^{i\omega})\Phi^*_n(e^{i\omega})]^{-1} .$$

The nth order minimum variance, $\mathbf{MV}_n(\omega)$ spectral estimator is given by

$$\mathbf{MV}_n(\omega)=\mathbf{M}_n^{-1}(\omega)=[\sum_{k=0}^{n} \mathbf{AR}_k^{-1}(\omega)]^{-1}=([z^{\sigma_0}\mathbf{I},z^{\sigma_1}\mathbf{I},\ldots,z^{\sigma_0}\mathbf{I}]\mathbf{R}_n^{-1}[\overline{z}^{\sigma_0}\mathbf{I},\overline{z}^{\sigma_1}\mathbf{I},\ldots,\overline{z}^{\sigma_0}\mathbf{I}])^{-1} \tag{2.1}$$

The main result of [1] demonstrated that $\mathbf{MV}_n(\omega)$ decreases monotonically to the two-sided point spectrum of the random field given by

$$\mathbf{P}_\mathbf{Y}(\omega) = \begin{cases} (\mathbf{A}_j\mathbf{A}_j^*)/4, & \omega_j = \pm\gamma_j \\ 0, & \text{otherwise} \end{cases} . \tag{2.2}$$

The method of [1] relies on the convergence of the $\mathbf{MV}_n(\omega)$ spectra as a means of identifying signal wavenumbers. However, in practice an infinite sequence of correlations is never available and so a convergence test must be performed on the finite order MV spectra. The convergence test of [4] for the random process setting is readily extended to the random field setting and will be used in this work to test for the presence of harmonic signals. The convergence test relies on a subsequence $\{\mathbf{M}_{m(n)}(\omega)\}_{n=0}^{\infty}$ of $\{\mathbf{M}_n(\omega)_{n=0}^{\infty}\}$. The motivation for this test stems from the fact that the $\mathbf{AR}_n$ spectra converge to the noise spectrum at all non-signal frequencies [8]. Noting that $\mathbf{M}_n(\omega)$ is a sum of n $\mathbf{AR}(\omega)$ estimates, then one can show that for any positive integer k_0 and sufficiently large n

$$\mathbf{M}_{k_0 n}(\omega) \approx k_0 \mathbf{M}_n(\omega) \tag{2.3}$$

at any noise frequency ω. Letting the subsequence index $m(n)=m_0 k_0^n$, then

$$\mathbf{M}_{m_0 k_0^{n+1}}(\omega) = k_0 \mathbf{M}_{m_0 k_0^n}(\omega), \tag{2.4}$$

where m_0 is chosen to be the order n beyond which the property (2.3) holds. Given a finite autocorrelation sequence and a value of m_0, the subsequence with the most elements uses $k_0=2$. In this case, for sufficiently large model orders the subsequence elements $\mathbf{M}_{m(n)}(\omega)$ and $\mathbf{MV}_{m(n)}(\omega)$ will double and halve, respectively, (since $\mathbf{MV}_n(\omega)=\mathbf{M}_n(\omega)^{-1}$), at every non-signal frequency as n is incremented (i.e. as $m(n)=m_0 2^n$ doubles). Using the above property a convergence test for identifying sinusoids is now presented.

Given $\mathbf{M}_{m(n)}(\omega)$ and m(n) a sequence of doubling model orders; $m(n)=m_0 2^n$, then

$$\lim[\mathbf{M}_{m(n)}(\omega)]^{1/(\log_2[m(n)])} = \left(\begin{array}{l} 1 \text{ for } \omega_j=\gamma_j \\ 2 \text{ for } \omega_j\neq\gamma_j \end{array}\right. \tag{2.5}$$

The choice of the exponent term $\log_2[m(n)]$ in (2.5) is related to the number of such subsequences, each beginning with a different value of m_0, which may be desired. The idea here is to utilize as many of the $\mathbf{MV}_n$ spectra as possible. By using this term, the convergence rates of the various subsequences will be relatively insensitive to small changes in m_0. For example, suppose $\mathbf{M}_n$ is available through order n=200 and that n=20 is sufficient to begin to observe the behaviour described by (2.4). Then the subsequences with orders {20,40,80,160} and {22,44,88,176} should converge to either 1 or 2 at similar rates. Note that a similar test using $\mathbf{MV}_{m(n)}(\omega)$, can be used, in which case (2.5) would converge to 1/2 at each non-signal frequency.

In order to accommodate settings involving large numbers of sinusoids an automated procedure which can ascertain (2.5) given a finite amount of correlation information is desirable. To this end, notice that for large n, the subsequence in (2.5) will follow the curve $Kx^{1/(\log 2[m(n)])}$, where theoretically K equals 1 or 2 [8]. This suggests specifying a threshold, K_{th}, such that if a least squares fit of the subsequence to this two parameter model yields $K<K_{th}$ then convergence of (2.5) to 1 is assumed; otherwise, convergence to 2 is assumed. An advantage of this curve fit is that only two subsequence values are necessary to obtain a value for K. Any frequency for which K satisfies $K<K_{th}$ is a potential sinusoidal frequency. Often, within the region of an actual harmonic signal component (sinusoid), there are several frequencies which satisfy $K<K_{th}$. In this case, the frequencies satisfying the inequality are checked for a local minima of K. The frequency corresponding to the local minima of K is considered to be the frequency of a harmonic signal component.

3. Efficient computation of $\mathbf{MV}_n(\omega)$.

Equation (2.1) shows two equivalent representations of the $\mathbf{MV}_n(\omega)$ spectral estimator. The lack of a m-D Levinson-type algorithm for recursively computing the **AR** spectra, forces the use of the latter representation involving the inverse of the correlation matrix $\mathbf{R}_n$ as our method of computing the spectrum. This method requires that $\mathbf{R}_n$ be inverted for each model order in the chosen subsequence m(n) so that each $\mathbf{MV}_n(w)$ spectrum can be computed. We now present an eigendecomposition-based method that efficiently computes these spectra. The symmetric correlation matrix $\mathbf{R}_n$ can be decomposed into the matrix product of its eigenvalues and eigenvectors as follows

$$\mathbf{R}_n = \mathbf{Q}_n\Lambda_n\mathbf{Q}_n^{-1}$$

where $\mathbf{Q}_n$ is the (n+1)×(n+1) matrix of column eigenvectors, $\mathbf{Q}_n^{-1}=\mathbf{Q}_n^t$, and Λ_n is the diagonal matrix of eigenvalues. Given this representation, $\mathbf{R}_n^{-1}$ can be expressed as

$$\mathbf{R}_n^{-1} = \mathbf{Q}_n\Lambda_n^{-1/2}\Lambda_n^{-1/2}\mathbf{Q}_n^t \tag{3.1}$$

Substituting (3.1) into (2.1) yields the $\mathbf{M}_n(\omega)$ estimate

$$\mathbf{M}_n(z) = [z^{\sigma_0}\mathbf{I}, z^{\sigma_1}\mathbf{I},\ldots,z^{\sigma_n}\mathbf{I}]\mathbf{Q}_n\Lambda_n^{-1/2}\Lambda_n^{-1/2}\mathbf{Q}_n^t[\overline{z}^{\sigma_0}\mathbf{I},\overline{z}^{\sigma_1}\mathbf{I},\ldots,\overline{z}^{\sigma_n}\mathbf{I}]^t \tag{3.2}$$

Note that multiplying $\mathbf{Q}_n=[q_0,q_1,..,q_n]^t$ by the vector $[z^{\sigma_0},z^{\sigma_1},\ldots,z^{\sigma_n}]$ is equivalent to taking an m-D DFT of each column eigenvector q_i. Note also that the last three factors in the matrix multiplication produce the conjugate transpose of the first three factors, i.e.

$$[z^{\sigma_0}\mathrm{I}, z^{\sigma_1}\mathrm{I},\ldots,z^{\sigma_n}\mathrm{I}]\mathbf{Q}_n\Lambda_n^{-1/2} = [\ \Lambda_n^{-1/2}\mathbf{Q}_n^t[\overline{z}^{\sigma_0}\mathrm{I},\overline{z}^{\sigma_1}\mathrm{I},\ldots,\overline{z}^{\sigma_n}\mathrm{I}]^t\ ]^*, \quad * \equiv \text{conjugate transpose} \tag{3.3}$$

This produces an alternative representation of $\mathrm{MV}_n(z)$

$$\begin{aligned}\mathbf{MV}_n(z) &= \{\sum_{i=0}^{n} \mathrm{DFT}\{\lambda_i^{-1/2}\ \mathbf{q}_i(\sigma)\} \cdot \overline{\mathrm{DFT}\{\lambda_i^{-1/2}\ \mathbf{q}_i(\sigma)\}}\ \}^{-1}, \qquad \overline{} \equiv \text{conjugate} \\ &= \{\mathrm{DFT}\{\ \sum_{i=0}^{n} \lambda_i^{-1}[\mathbf{q}_i(\sigma) \star \mathbf{q}_i(-\sigma)]\ \}\}^{-1}, \quad \star \equiv \text{convolution}\end{aligned} \tag{3.4}$$

where we make use of the transform theory result that a multiplication in the frequency domain is equivalent to a convolution in the spatial domain. Note that this final form involves only a single m-D DFT of a sum of convolved sequences and is far more efficient than taking an m-D DFT of each column of $\mathbf{Q}_n$.

4. Example

We now present an example to illustrate the concepts discussed in this paper and to show that the $\mathrm{MV}_n(\omega)$ spectrum approach has notable advantages for identifying sinusoids in the random field setting. The single channel random field consists of two closely spaced sinusoids within a 2D resonant random noise field, where the signal frequencies are close to the resonant frequency of the random noise field. The random field data was generated for the following array
where $0<n\leq100$, and $0<m\leq100$ and w(n,m) is a colored random noise field whose theoretical

$$x(m,n) = 2\cos(16\pi/32n + 16\pi/32m) + \cos(12\pi/32n + 16\pi/32m) + w(n,m)$$

frequency spectrum is shown in Figure 1. The noise field consists of a resonant peak at the frequency (16π/32,16π/32). Note in this figure and for all frequency spectrum plots, only the 1st quadrant of frequencies is shown since it contains all of the relevant information for this example. Notice at signal frequency 1, (16π/32,16π/32), the noise power is approximately 9dB and hence the local signal-to-noise-ratio (SNR) is about -9dB. At signal frequency 2, (13π/32,16π/32), the noise power is approximately 5dB yielding a local SNR of roughly -11dB.

The MV spectra were computed for orders 16, 32, 64, 128, 256 and 512 based upon the lagged product correlation sequence estimate computed from a single observation of the above mentioned data array. The DFT and AR spectra were also computed for this same data array as a means for comparison.

The MV spectra for orders 256 and 512 appear in Figure 2. From these plots, it is evident that at all non-signal frequencies the spectrum is being driven down by approximately 3dB for the doubling of the order of the MV estimate. Furthermore, at the two signal frequencies, the MV estimate appears to be converging, as the drop in dB due to doubling the model order is very small. It is precisely this monotonic convergence of the MV spectra at signal frequencies that allows for the accurate detection of sinusoidal signals. The convergence test for K_{th}=1.5 was applied to the MV spectral data resulting in identification of both sinusoids only. The estimated K value for frequency 1 is 1.035 and for frequency 2 is 1.083. Both are close to the theoretical value of 1.0. It is worth mentioning that the convergence test used to identify sinusoidal frequencies based on a family of MV spectra is entirely automated such that there is actually no need for visual inspection of the spectra. Figure 3 shows a single row of the frequency grid for the 6 computed MV spectra. The y-axis frequency is fixed at f_y=16π/32, or frequency bin 32 for a 128 point spectrum. This frequency is common to both the signal frequencies and clearly illustrates the convergence patterns of the two signals. Notice that the signal frequency (16π/32,16π/32) converges more quickly than does the second signal frequency (13π/32,16π/32) and this can be explained by the fact that the second frequency has a lower SNR and so higher orders of MV spectra are required to sufficiently drive out the noise.

To further illustrate the convergence of the MV spectra at signal frequencies and the divergence of the spectra at noise frequencies, the MV spectra dB values for all 6 spectra have been plotted in Figure 4 for both a signal frequency and a noise frequency. It is obvious from the noise plot that doubling the order of the MV estimate causes the spectrum to drop by 3 dB with each iteration as the plot is roughly a straight line with a -3dB slope. On the other hand, the signal plot shows that the dB drop for successive iterations is much smaller and more importantly is decreasing each time.

Returning to Figure 3, it is apparent from the plots that at frequency 1, (16π/32,16π/32), the spectra are converging to approximately 0dB which is precisely $A_1A_1^*/4$ as indicated by (2.2) for A_1=2. Similarly, the spectra at frequency 2, (13π/32,16π/32), are converging to approximately -6dB, which is also in accordance with (2.2) for A_2=1. Therefore, the MV spectra are indeed converging to the two sided point spectrum of the random field. Note at order 512, the MV spectral estimate is about 0.2dB instead of 0dB for frequency 1 and -5.6dB instead of -6dB for frequency 2. This is due to the fact that the order is finite and full convergence has not been reached, so the MV estimate of highest order will be a conservative estimate of the power.

Figures 5 and 6 show the AR_{256} and DFT spectral estimates of the data sequence. It is important to note that the AR estimate contains several peaks in the spectrum that could be interpreted as sinusoidal sources if no a priori information is known about the process.

This phenomenon is precisely the spectral leakage problem discussed in section 1 that is commonplace in AR estimates. It is apparent that the MV spectral estimator did not suffer from spectral leakage as no false signal frequencies are detected and power is determined by the peak height and not by integration or summation under the peak. Observing the DFT spectral estimate shows that this same problem is considerably worse. This illustrates how the problem of spectral leakage which is evident in the 1-D setting for both of these estimators, is further compounded when extending to the 2-D setting. Furthermore, for both the AR and the DFT, there is no basis for associating any given spectral peak with a sinusoid as opposed to a narrowband spectral noise component such as a resonance.

5. Discussion and Summary of Results

The purpose of this paper is to demonstrate the enhanced utility of a MV spectrum-based technique for identifying a harmonic signal field. The enhanced utility of our technique is not unexpected since the MV spectral estimator can be expressed as an optimal approach based on Hilbert space projections [1]. The technique can identify point spectrum associated with harmonic signal components in unknown, arbitrary, and possibly nonstationary random noise. It derives its power in a novel way by using the monotonic convergence properties of a sequence of MV spectra. By observing a sequence of MV spectra, as opposed to a single spectra, it is possible to discern between sinusoidal signals and narrowband random noise components. Specifically, pure random noise frequencies are found to converge in the limit at the rate of 3dB per doubling of model order. These strong convergence properties allow for the use of an automated convergence test which can identify sinusoidal signals without the need for visual inspection of the spectra. From a practical view point, noise is driven out to the point where any remaining harmonic signal components are considered insignificant due to their small power. These results are in stark contrast to the m-D DFT and m-D polynomial techniques previously discussed. These techniques only utilize a single spectra, they do not have comparable convergence properties, and they require a priori knowledge to definitively distinguish harmonic signal components from narrowband random noise.

The proposed technique yields easy to interpret, accurate, and conservative estimates of the power for each sinusoidal signal. Since the MV spectrum is a true power spectrum, power estimates are obtained directly from the spectral height at a given frequency. Once again this is in stark contrast to the m-D DFT and m-D polynomial techniques which estimate power by integration or summation under a peak. Hence, the MV spectrum does not have the spectral leakage associated with the other techniques.

The effectiveness of our proposed technique is actually enhanced as the number of dimensions is increased. To see this point, consider that our proposed technique relies on the computation of the MV spectra for doubled model orders. In this context, the final MV spectra model order can be quite large. Therefore, for a single dimension, a 1024^{th} order MV spectrum would require a reliable estimate of the system correlation sequence through the 1024^{th} correlation lag. Reliability for such a large lag can be hard to achieve. However, for two dimensions, a 1024^{th} MV spectrum only requires a reliable correlation sequence estimate through the 32^{nd} correlation lag in both dimensions, since $32 \times 32 = 1024$. Hence, the technique's effectiveness is enhanced as the dimension is increased since large model orders can be achieved using only relatively low correlation lags.

6. References

1. A.E.Frazho and P.J.Sherman 1991, IEEE Trans. on Info. Th. **37(9)**, pp. 1457-1459. On the convergence of the minimum variance spectral estimator in nonstationary noise.
2. L.S.Marple 1987 Digital Spectral Analysis with Applications. Prentice Hall.

3. C.Foias, A.E.Frazho, and P.J.Sherman 1990 Mathematics of Control, Signals and Systems, **3**, pp.31-43. An new approach for determining the spectral data of multichannel harmonic signals in noise.
4. D.E.Lyon and P.J.Sherman 1993 IEEE Trans. on Acoustics, Speech, and Signal Processing, **41(11)**, pp.3177-3182. Practical issues concerning the family of multichannel MV spectra for recovery of point spectrum.
5. P.J.Sherman and K.N.Lou 1991 IEEE Transactions on Acoustics, Speech and Signal Processing, **39(11)**, pp.644-655. On the family of ML spectral estimates for mixed spectrum identification.
6. K.N.Lou, P.J.Sherman, and D.E.Lyon 1993 Mechanical Systems and Signal Processing, 7(1), pp.13-27. System identification and coherence analysis in the presence of a harmonic signal.
7. D.E.Brown and P.J.Sherman 1990 Proc. 10th Intl. Compressor Conf., Purdue University, West Lafayette, IN, July 17-20, 1990, pp.625-633. The utility of three multichannel spectral techniques for multipiston compressor noise source identification.
8. D.E. Lyon 1990 Ph.D. dissertation in Mechanical Eng. at Purdue University. Multidimensional maximum likelihood spectral estimation for periodically excited mechanical systems.

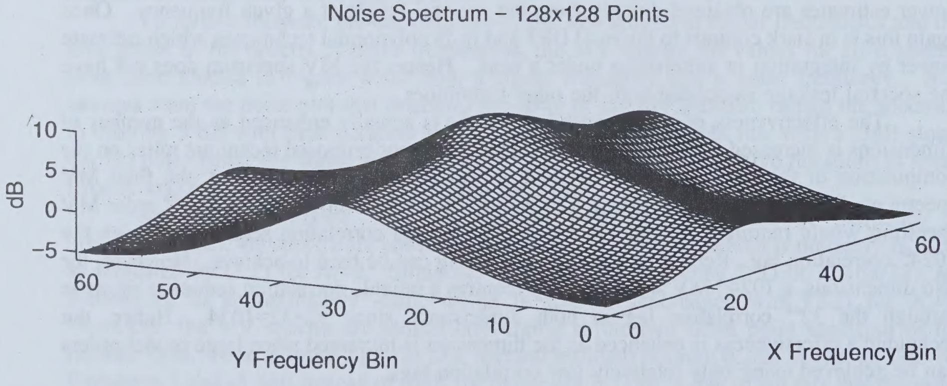


FIG. 1. Theoretical Frequency Spectrum of Random Noise Field.

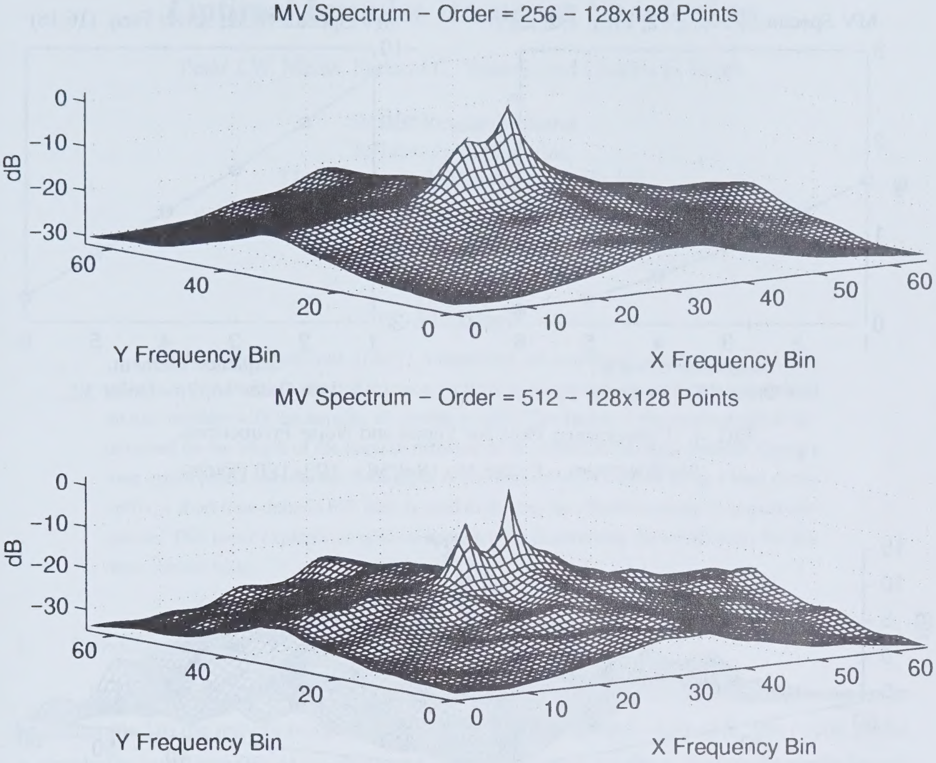


FIG. 2. MV Spectra of Random Field.

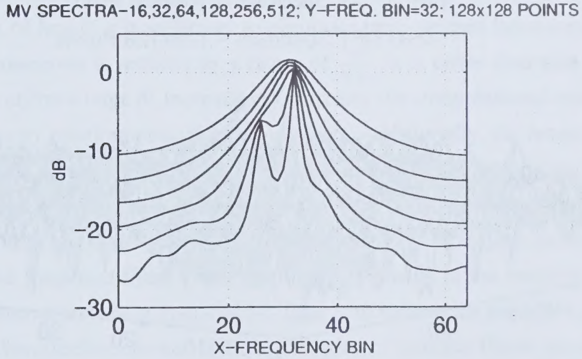
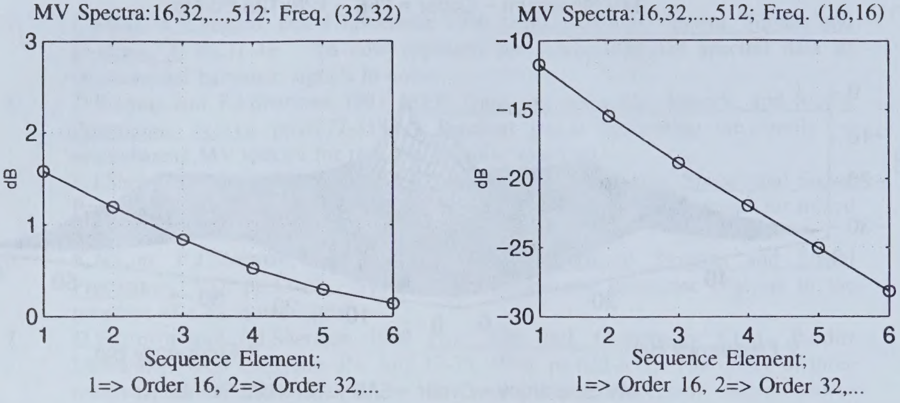


FIG. 3. MV Spectra for Y-Frequency = $16\pi/32$.


MV Spectra:16,32,...,512; Freq. (32,32)

MV Spectra:16,32,...,512; Freq. (16,16)

FIG. 4. Convergence Plots for Signal and Noise Frequencies.

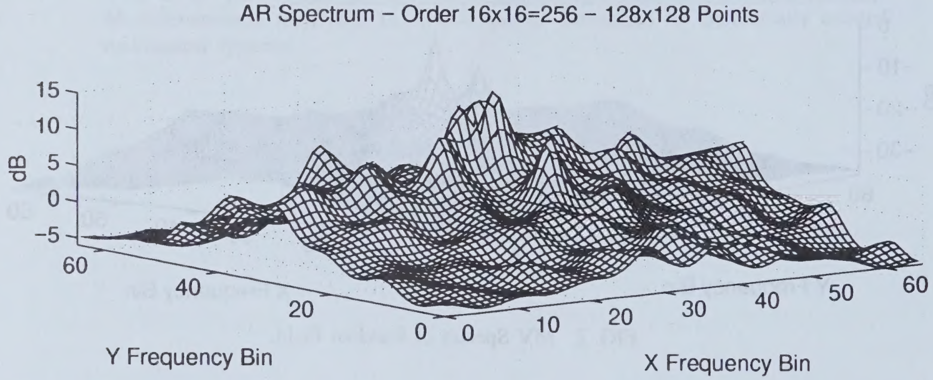

AR Spectrum − Order 16x16=256 − 128x128 Points

FIG. 5. AR Spectrum of Random Field.

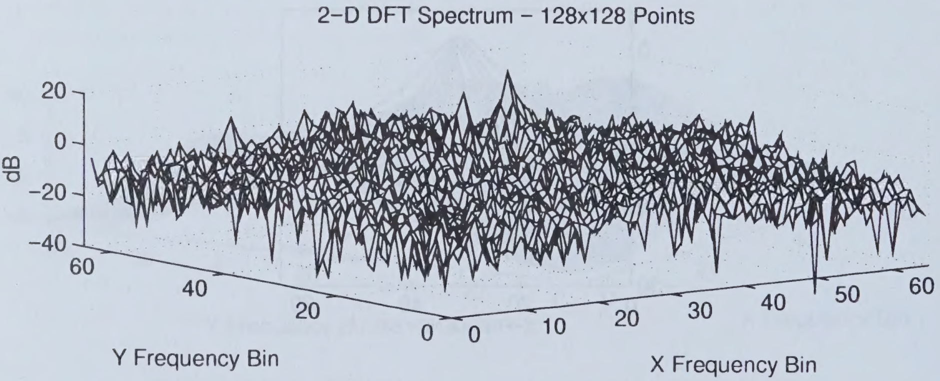

2−D DFT Spectrum − 128x128 Points

FIG. 6. DFT Spectrum of Random Field.

Optimal Impulse Response Shortening

Peter J.W. Melsa, Richard C. Younce, and Charles E. Rohrs

Tellabs Research Center
Tellabs Operations, Inc.
3740 Edison Lakes Parkway, Suite 101
Mishawaka, IN 46545
(219)259-9818

Abstract

In Discrete MultiTone (DMT) transceivers an intelligent guard time sequence, called a cyclic prefix, is inserted between symbols to ensure that samples from one symbol do not interfere with the samples of another symbol. The length of the cyclic prefix is determined by the length of the impulse response of the effective physical channel. Using a long cyclic prefix reduces the throughput of the transceiver. To avoid using a long cyclic prefix, a short time-domain FIR filter is used to shorten the effective channel's impulse response. This paper explores an optimal approach for determining the coefficients for this time-domain filter.

I. *Introduction*

In Discrete MultiTone (DMT) transceivers each symbol is comprised of N samples to be transmitted to the remote receiver plus a cyclic prefix of length v (see [1]). The cyclic prefix is simply the last v samples of the original N samples to be transmitted. The cyclic prefix length is determined by the length of the channel's impulse response, and is chosen to minimize inter-symbol interference. At the receiver the cyclic prefix is discarded, the remaining N samples are then processed by the receiver. If the impulse response of the channel is of length $v+1$ or shorter then a cyclic prefix of length v is sufficient to eliminate inter-symbol interference. Since the efficiency of the transceiver is reduced by a factor of $\frac{N}{N+v}$ it is either desirable to make v as small as possible or utilize a large N. Increasing N increases the computational complexity, system delay, and memory requirements of the transceiver. Additionally, the length of the channel's impulse response is typically not under the control of the designer varying from channel to channel, leaving the designer to choose both a large v and large N in order to achieve reasonable efficiency. To alleviate these problems a short time-domain FIR filter, referred to here as a Shortened Impulse Response Filter (SIRF), is typically placed in the receiver immediately following the A/D converter. The purpose of this filter is to shorten the impulse response of the effective channel. The effective channel is the convolution of transmit filters, physical channel, receive filters, and the SIRF. The impulse response of the effective channel needs to be shorter than the length of the cyclic prefix. The length of the SIRF is usually fixed *a priori* and not changed from channel to channel.

This paper addresses the calculation of the coefficients for the Shortened Impulse Response Filter (SIRF). It was shown in [2] that the greatest throughput, for a given computational complexity, of a DMT transceiver can be accomplished using an equalizer that shortens the impulse response of the channel and a cyclic prefix. This approach was demonstrated for a DMT based HDSL transceiver in [1]. In [3] an efficient means for determining the coefficients of the SIRF was developed. However, the algorithms in [3] are hampered by stability and convergence problem. This paper develops an optimal approach to impulse response shortening.

Section II.A develops the motivation for channel impulse response shortening for DMT transceivers. Section II.B presents a new algorithm based on optimal shortening of an impulse response. Given the original impulse response, a SIRF length, and a cyclic prefix length the optimal algorithm finds the coefficients which generate the shortest impulse response. Finally, some simulation results are reported in Section III, and conclusions are drawn in Section IV.

II. Channel Shortening

II.A. Background

A simplified DMT transceiver (without echo cancellation) is shown in Figure 1. The encoder maps incoming bits onto $\frac{N}{2}+1$ carrier frequencies forming the complex frequency domain subsymbols $X(k)$ where $k=0,1,\ldots\frac{N}{2}$. The $\frac{N}{2}+1$ subsymbols taken together form a frequency domain symbol. A conjugate symmetry condition is then imposed creating a length N frequency domain signal. This signal is inverse Fourier transformed to generate a real, length N time domain symbol, $x(n)$, which is prefixed with a Cyclic Prefix (CP) of length v and transmitted to the far end receiver. The CP consists of the last v samples of $x(n)$. At the receiver, the receive signal is passed through a short time-domain FIR filter, and then the CP is removed. This signal, $y(n)$ is then Fourier transformed back to the frequency domain. Each of the frequency domain subsymbols $Y(k)$ can then be decoded to an outgoing bit stream.

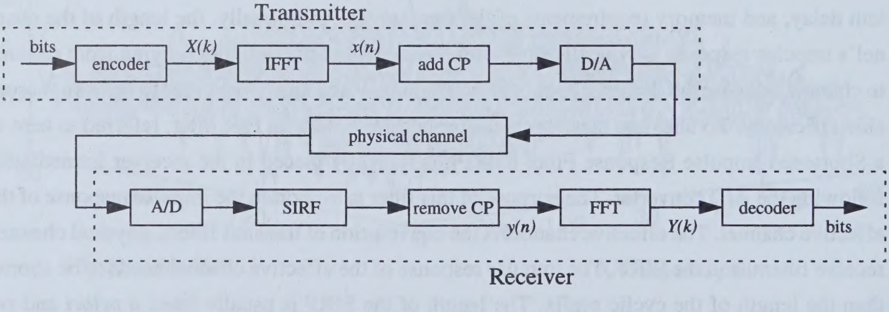


Figure 1: Simplified DMT Transceiver

In Figure 1 the physical channel is considered to be the combination of the actual cable and any filtering done in the transmit or receive analog circuitry. Denoting the impulse response of the physical channel by $h(n)$, the output of the SIRF can be expressed as

$$y(n) = (h(n) * w(n)) * x(n) = h_{eff}(n) * x(n) , \tag{1}$$

where $w(n)$ is the impulse response of the SIRF and $*$ denotes the convolution operator. It can be shown that if $x(n)$ is transmitted with a cyclic prefix of length v and the effective channel, $h_{eff}(n)$, is at most of length $v+1$, each $y(n)$ in the current symbol will only depend upon $x(n)$'s currently being transmitted. If the effective channel is not constrained to a length of $v+1$, then $y(n)$ will depend upon the current symbol's time samples $x(n)$ as well as the previous symbol's time samples $x_p(n)$. The previous transmit symbol $x_p(n)$ will then contribute inter-symbol interference which presents itself as noise to the slicer and so decreases the performance of the transceiver.

Regardless of the choice of $w(n)$ it is generally not possible to shorten the impulse response perfectly. Some energy will lie outside the largest $v+1$ consecutive samples of $h_{eff}(n)$. As a measure of the inter-symbol interference we can measure the shortening SNR (SSNR) which is the ratio of the energy in the largest consecutive $v+1$ samples to the energy in the remaining samples. The largest $v+1$ samples will not necessarily start with the first sample. This delay, d, is normally compensated for at the receiver by delaying the start of the receive symbol. Since $h(n)$ is not under the control of the designer, we use the coefficients of the SIRF, $w(n)$, to limit the length of the effective channel's impulse response, thereby minimizing inter-symbol interference. The shape of the resulting impulse response of the effective channel, $h_{eff}(n)$, is usually unimportant, what is important is that the *SSNR* be maximized.

A necessary step in computing SIRF coefficients is to estimate the impulse response of the channel. Typically, DMT transceivers use known training sequences for purposes of adapting various parameters associated with the transmitter and receiver. Since DMT transceivers have the frequency response naturally available, the training sequences can be used to calculate the frequency response of the channel. The frequency response can then be inverse transformed into the time domain, generating the time-domain impulse response. Care should be taken to reduce the variance of the noise in the estimate, as well as compensate for any other known effects such as aliasing.

II.B. Optimal Shortening

In this section we develop the notion of optimal shortening of an impulse response. The algorithm utilizes eigenvalues and eigenvectors to generate the coefficients of an optimal shortening filter, given the original impulse response, the CP length and the SIRF length.

We can rewrite Eq. (1) using matrices as

$$\boldsymbol{h}_{eff} = \boldsymbol{H}\boldsymbol{w} = \begin{bmatrix} h_{eff}(0) \\ h_{eff}(1) \\ \vdots \\ h_{eff}(M-1) \\ h_{eff}(M) \\ \vdots \\ h_{eff}(M+t-2) \end{bmatrix} = \begin{bmatrix} h(0) & 0 & \cdots & & 0 \\ h(1) & h(0) & \ddots & & \\ \vdots & \vdots & & & \\ h(M-1) & h(M-2) & \cdots & h(M-t+1) & h(M-t) \\ 0 & h(M-1) & h(M-2) & \cdots & h(M-t+1) \\ \vdots & & \ddots & & \vdots \\ 0 & \cdots & & 0 & h(M-1) \end{bmatrix} \begin{bmatrix} w(0) \\ w(1) \\ \vdots \\ w(t-1) \end{bmatrix}, \quad (2)$$

where *M* is the length of the physical channel's impulse response and *t* is the length of the SIRF. We wish to force as much of the effective channel's impulse response to lie in *v*+1 consecutive samples and as a consequence minimize inter-symbol interference. Let $\boldsymbol{h}_{win}$ represent a window of *v*+1 consecutive samples of $\boldsymbol{h}_{eff}$ starting with sample *d*, and let $\boldsymbol{h}_{wall}$ represent the remaining *M+t+v-2* samples of $\boldsymbol{h}_{eff}$. With these definitions

$$\boldsymbol{h}_{win} = \begin{bmatrix} h_{eff}(d) \\ h_{eff}(d+1) \\ \vdots \\ h_{eff}(d+v) \end{bmatrix} = \begin{bmatrix} h(d) & h(d-1) & \cdots & h(d-t+1) \\ h(d+1) & h(d) & \cdots & h(d-t+2) \\ \vdots & & \ddots & \vdots \\ h(d+v) & h(d+v-1) & \cdots & h(d+v-t+1) \end{bmatrix} \begin{bmatrix} w(0) \\ w(1) \\ \vdots \\ w(t-1) \end{bmatrix} \equiv \boldsymbol{H}_{win}\boldsymbol{w} \quad (3)$$

$$\boldsymbol{h}_{wall} = \begin{bmatrix} h_{eff}(0) \\ \vdots \\ h_{eff}(d-1) \\ h_{eff}(d+v+1) \\ \vdots \\ h_{eff}(M+t-2) \end{bmatrix} = \begin{bmatrix} h(0) & 0 & \cdots & 0 \\ \vdots & \ddots & & \\ h(d-1) & h(d-2) & \cdots & h(d-t) \\ h(d+v+1) & h(d+v) & \cdots & h(d+v-t+2) \\ \vdots & & \ddots & \\ 0 & \cdots & 0 & h(M-1) \end{bmatrix} \begin{bmatrix} w(0) \\ w(1) \\ \vdots \\ w(t-1) \end{bmatrix} \equiv \boldsymbol{H}_{wall}\boldsymbol{w} \ . \quad (4)$$

Optimal shortening can be expressed as $\min_{w} \boldsymbol{h}^T_{wall}\boldsymbol{h}_{wall}$ while satisfying the constraint: $\boldsymbol{h}^T_{win}\boldsymbol{h}_{win} = 1$. Constraining the energy in the window ensures that the trivial solution, $\boldsymbol{w} = [0 \ldots 0]^T$, is disallowed. The value of 1 is picked arbitrarily since $\boldsymbol{w}$ can be scaled arbitrarily without changing the energy ratio. The expressions for the energy outside and inside the window can be written as

$$\boldsymbol{h}^T_{wall}\,\boldsymbol{h}_{wall} = \boldsymbol{w}^T\boldsymbol{H}^T_{wall}\,\boldsymbol{H}_{wall}\boldsymbol{w} = \boldsymbol{w}^T\boldsymbol{A}\,\boldsymbol{w} \quad (5)$$

$$\boldsymbol{h}^T_{win}\,\boldsymbol{h}_{win} = \boldsymbol{w}^T\boldsymbol{H}^T_{win}\,\boldsymbol{H}_{win}\boldsymbol{w} = \boldsymbol{w}^T\boldsymbol{B}\,\boldsymbol{w}\ , \quad (6)$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ are symmetric and positive semidefinite matrices. Optimal shortening can be considered as choosing $\boldsymbol{w}$ to minimize $\boldsymbol{w}^T\boldsymbol{A}\boldsymbol{w}$ while satisfying a constraint of $\boldsymbol{w}^T\boldsymbol{B}\boldsymbol{w}=1$.

The following development assumes that $\boldsymbol{B}$ is invertible. The case where $\boldsymbol{B}$ is singular is more complex and is handled in Appendix A. Typical DMT transceivers use a SIRF length which is shorter than the length of the CP (i.e. $t<v$). The rows of $\boldsymbol{H}_{win}$ consist of shifted windows of the channel impulse response and it has been found in practical applications (e.g. twisted-pair copper cables) that the rank($\boldsymbol{H}_{win}$)=t. As a consequence, the matrix $\boldsymbol{B}$ of dimension t x t will be positive definite and can be decomposed using Cholesky Decomposition into

$$\boldsymbol{B} = \boldsymbol{Q}\Lambda\boldsymbol{Q}^T = (\boldsymbol{Q}\sqrt{\Lambda})(\sqrt{\Lambda}\boldsymbol{Q}^T) = (\boldsymbol{Q}\sqrt{\Lambda})(\boldsymbol{Q}\sqrt{\Lambda})^T = \sqrt{\boldsymbol{B}}\sqrt{\boldsymbol{B}}^T, \tag{7}$$

where Λ is a diagonal matrix formed from the eigenvalues of $\boldsymbol{B}$, and the columns of $\boldsymbol{Q}$ are the orthonormal eigenvectors. Note that, since $\boldsymbol{B}$ is of full rank, the matrix $(\sqrt{\boldsymbol{B}})^{-1}$ exists.

In order to satisfy the constraint on $\boldsymbol{w}^T\boldsymbol{B}\boldsymbol{w}$ define

$$\boldsymbol{y} = \sqrt{\boldsymbol{B}}^T\boldsymbol{w} \tag{8}$$

so that

$$\boldsymbol{y}^T\boldsymbol{y} = \boldsymbol{w}^T\sqrt{\boldsymbol{B}}\sqrt{\boldsymbol{B}}^T\boldsymbol{w} = \boldsymbol{w}^T\boldsymbol{B}\boldsymbol{w} = 1\,. \tag{9}$$

Solving in Eq. (8) for $\boldsymbol{w}$

$$\boldsymbol{w} = \left(\sqrt{\boldsymbol{B}}^T\right)^{-1}\boldsymbol{y} \tag{10}$$

we find that

$$\boldsymbol{w}^T\boldsymbol{A}\boldsymbol{w} = \boldsymbol{y}^T(\sqrt{\boldsymbol{B}})^{-1}\boldsymbol{A}(\sqrt{\boldsymbol{B}^T})^{-1}\boldsymbol{y} = \boldsymbol{y}^T\boldsymbol{C}\boldsymbol{y}, \tag{11}$$

where $\boldsymbol{C}$ is appropriately define. Optimal shortening can thus be considered as $\min_{y} \boldsymbol{y}^T\boldsymbol{C}\boldsymbol{y}$ while constraining $\boldsymbol{y}^T\boldsymbol{y} = 1$. The solution to this problem occurs for $\boldsymbol{y} = \boldsymbol{l}_{min}$ where $\boldsymbol{l}_{min}$ is the unit-length eigenvector corresponding to the minimum eigenvalue λ_{min} of $\boldsymbol{C}$. The resulting SIRF coefficients are thus

$$\boldsymbol{w}_{opt} = (\sqrt{\boldsymbol{B}^T})^{-1}\boldsymbol{l}_{min}\,. \tag{12}$$

Substituting Eq. (7) into Eq. (11) we find

$$\boldsymbol{C} = (\boldsymbol{Q}\sqrt{\Lambda})^{-1}\boldsymbol{A}(\sqrt{\Lambda}\boldsymbol{Q}^T)^{-1}\,. \tag{13}$$

The shortening SNR can be expressed as

$$SSNR_{opt} = 10\log\left(\frac{\boldsymbol{w}_{opt}^T\boldsymbol{B}\boldsymbol{w}_{opt}}{\boldsymbol{w}_{opt}^T\boldsymbol{A}\boldsymbol{w}_{opt}}\right) = 10\log\left(\frac{1}{\lambda_{min}}\right) = -10\log(\lambda_{min})\,. \tag{14}$$

In summary, the optimal algorithm uses a composite matrix, $\boldsymbol{C}$, whose elements are calculated using the wall and window impulse response matrices, $\boldsymbol{H}_{wall}$ and $\boldsymbol{H}_{win}$. The optimal SSNR is directly related to the minimum eigenvalue of this composite matrix and the optimal SIRF coefficients are a linearly transformed version of the unit-length eigenvector associated with the minimum eigenvalue.

The algorithm presented above required the choice of a window location, *i.e.* choice of d, on the impulse response prior to calculating the SIRF coefficients. This can be accomplished by trying different window locations, computing the SIRF coefficients, calculating the SSNR, and choosing the window location and SIRF coefficients which yielded the best SSNR. This can be very computational expensive and wasteful. Often the channel has an impulse response which is very short prior to the peak (pre-cursor) and very long after the peak (post-cursor). As such, the impulse response can often be considered as a few zeros creating the pre-cursor response and poles and zeros creating the post-cursor response. Taking advantage of this fact can reduce the computational complexity of coefficient calculation. It has been observed that one can achieve adequate shortening by choosing a window location based on the peak of the original impulse response. Only a minimal reduction in achieved shortening results, and the iterations on window placement are eliminated.

III. Simulation Results

To demonstrate the capabilities of the optimal algorithm presented above the configuration shown in Figure 2 is used. The physical cable used is 9 kft of 26 gauge unshielded twisted pair, and the bandwidth used is 20-900 kHz. This configuration is representative of that encountered in the central office (ATU-C) to remote terminal (ATU-RT) path in the Asymmetrical Digital Subscriber Lines (ADSL) environment [4]. The impulse responses for this configuration were provided for us by W.Y. Chen of Bellcore [5]. For the simulation we used a SIRF length of 16 (t=16) and a cyclic prefix of length 32 (v=32). A full search algorithm was used to optimize the location of the modeling window location.

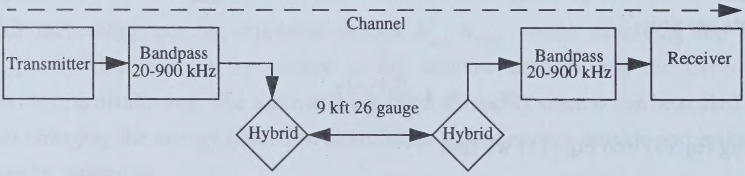


Figure 2: Configuration for Simulation Example

Figure 3 shows the original impulse response of the channel and the effective shortened channel if the optimal algorithm presented in Section II.B is used. The original impulse response has an SSNR=5.0 dB, while the optimal algorithm achieves a SSNR=64.1 dB. As a comparison, a more traditional least-squares approach [6] was used resulting in a SSNR=59.9 dB.

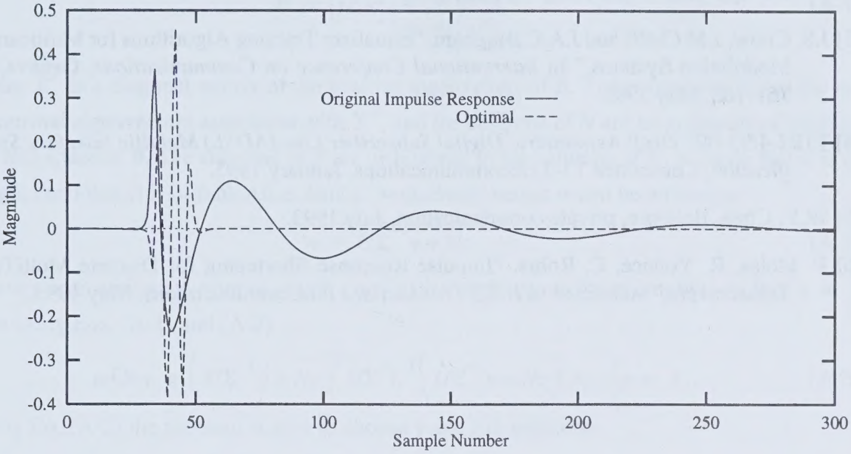


Figure 3: Original Channel Response & Optimally Shortened Effective Response

IV. Conclusions

In this paper, we have developed an optimal algorithm for computing the coefficients of the shortening impulse response filter (SIRF). Simulation results were provided to demonstrate the capabilities of the algorithm as well as compare it to a least-squares approach. There is also a great deal of promise in applying the optimal approach presented here to the situation where an echo path also exists in a DMT transceiver. The complexity of a DMT echo canceller depends upon the length of the echo impulse response. A "joint" optimal approach could then be used to simultaneously shorten both the echo and channel impulse responses, thereby reducing ISI and minimizing echo canceller complexity.

The algorithm presented here approached the problem of channel equalization from the perspective of impulse response shortening. It may be possible the SSNR measure is not the best measure of equalization, since ultimately the channel SNR in each sub-symbol of the multitone system is what is important. Future work may seek to develop SIRF coefficient algorithms which maximize the capacity of the multitone transceiver based upon the channel SNR. Additionally, it may be possible to apply the methods presented in this paper to communication systems that use modulation techniques other than multitone.

References

[1] J.S. Chow, J.C Tu and J.M. Cioffi, "A Discrete Multitone Transceiver System for HDSL Applications," *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 6, August 1991.

[2] J.S. Chow and J.M Cioffi, "A Cost-Effective Maximum Likelihood Receiver for Multicarrier Systems," *Proceedings of ICC*, 1992.

[3] J.S. Chow, J.M Cioffi and J.A.C Bingham, "Equalizer Training Algorithms for Multicarrier Modulation Systems," In *International Conference on Communications*, Geneva, pp 761-765, May 1993.

[4] T1E1.4/95-007 *Draft Asymmetric Digital Subscriber Line (ADSL) Metallic Interface Specification*, Committee T1-Telecommunications, January 1995.

[5] W.Y. Chen, Bellcore, private communication, July 1993.

[6] P. Melsa, R. Younce, C. Rohrs, "Impulse Response Shortening for Discrete MultiTone Transceivers," submitted to *IEEE Transactions on Communications*, May 1995.

Appendix A

As in Section II.B we desire to choose w to minimize w^TAw given $w^TBw=1$. Of particular interest here is the case where B is not invertible. Since B is positive semidefinite it can be written as

$$B = \begin{bmatrix} U \vdots N \end{bmatrix} \begin{bmatrix} \Sigma^2 & \vdots & 0 \\ \cdots & & \cdots \\ 0 & \vdots & 0 \end{bmatrix} \begin{bmatrix} U^T \\ \cdots \\ N^T \end{bmatrix} \tag{A-1}$$

where Σ^2 is a diagonal matrix of the positive eigenvalues of B. The columns of U are the orthonormal eigenvectors associated with Σ^2, and the columns of N are an orthonormal basis of the null space of B. The columns of N are orthogonal to the columns of U. Assume that w is of length t and that B is of rank r (i.e. $\dim(\Sigma^2)=r$). Every vector w can be written as

$$w = U\Sigma^{-1}y + Nz \tag{A-2}$$

where y is an arbitrary vector of length r and z is an arbitrary vector of length $(t\text{-}r)$. If $y^Ty = 1$ then using Eqs. (A-1) and (A-2)

$$w^TBw = \left(U\Sigma^{-1}y + Nz\right)^T U\Sigma^2U^T\left(U\Sigma^{-1}y + Nz\right) = y^Ty = 1 \tag{A-3}$$

Using Eq. (A-2) the problem is now to choose y and z to minimize

$$\left(y^T\Sigma^{-1}U^T + z^TN^T\right)A\left(U\Sigma^{-1}y + Nz\right) \tag{A-4}$$

given $y^Ty = 1$. First minimize with respect to z. Eq. (A-4) can be rewritten as

$$y^T\Sigma^{-1}U^TAU\Sigma^{-1}y + 2y^T\Sigma^{-1}U^TANz + z^TN^TANz. \tag{A-5}$$

Setting the derivative of Eq. (A-5) with respect to z equal to zero, we get

or
$$z = -\left(N^TAN\right)^{-1}N^TAU\Sigma^{-1}y \tag{A-6}$$

Substituting Eq. (A-6) into Eq. (A-4), the quantity that is left to be minimized is

$$\left(y^T\Sigma^{-1}U^T - y^T\Sigma^{-1}U^TAN\left(\left(N^TAN\right)^{-1}\right)^T\right) A\left(U\Sigma^{-1}y - N\left(N^TAN\right)^{-1}N^TAU\Sigma^{-1}y\right) = y^TCy \tag{A-7}$$

where C is appropriately define. Using Eqs. (A-2) and (A-6) the resulting optimal SIRF coefficients are

$$w_{opt} = \left(I - N\left(N^TAN\right)^{-1}N^TA\right)U\Sigma^{-1}l_{min}, \tag{A-8}$$

where l_{min} is the unit-length eigenvector associated with the minimum eigenvalue of C given in Eq. (A-7).

Parallel Matrix Sign Iterations*

Bertan Bakkaloğlu and Çetin Kaya Koç
Department of Electrical & Computer Engineering
Oregon State University
Corvallis, Oregon 97331

Abstract

We have implemented three matrix sign function algorithms on a Meiko CS-2 multiprocessor using the Parallel Virtual Machine (PVM) software. These matrix sign function algorithms are the well-known Newton's method [4], the partial fraction expansion algoritm [12], and the continued fraction algorithm [3]. The parallelization of Newton's method is straightforward since it utilizes parallel LU decomposition. The parallel matrix sign function algorithm based on partial fraction expansion was given by Pandey, Kenney, and Laub in [12]. Our contribution in this paper was to introduce a new parallel iterative algorithm to compute the sign function of a complex matrix, based on the continued fraction expansion of the inverse of the principal square root function. This algorithm uses parallel matrix mutiplications and linear system solutions at each step, and is suitable for vector and parallel machines due to its multiplication-rich nature. We have written all three algorithms in PVM, obtaining a highly portable code which can be compiled and executed on a network of workstations as well as on sophisticated multiprocessor machines. We summarize the results of our experiments on an 8-processor Meiko CS-2.

1 Introduction

The matrix sign function has several applications in system theory and matrix analysis, including solution of algebraic Riccati and matrix Lyapunov equations [10], system decomposition, model reduction [11], separation of eigenpairs [5], condition theory [9], and most recently for the numerical solution of M/G/1 and G/M/1 type Markov chains [1]. The matrix sign function maps the stable and unstable eigenvalues of a given matrix to -1 and 1, respectively, while preserving the eigenvectors of the original matrix. This property of the matrix sign function is useful for studying the eigenstructures of matrices without explicitly computing the eigenvalues. The sign of a complex scalar λ is defined over $\mathrm{Re}(\lambda) \neq 0$ by

$$\mathrm{sign}(\lambda) = \begin{cases} 1 & \text{if } \mathrm{Re}(\lambda) > 0 \ , \\ -1 & \text{if } \mathrm{Re}(\lambda) < 0 \ . \end{cases}$$

This definition can be extended to a matrix $A \in \mathcal{C}^{n \times n}$ whose eigenvalues do not lie on the imaginary axis. Let M take A to its Jordan form J as

$$A = MJM^{-1}. \tag{1}$$

---

*This research was supported in part by the National Science Foundation under the grants ECS–9312240 and CDA–9216172.

Let J be defined as

$$J = \begin{bmatrix} J_+ & 0 \\ 0 & J_- \end{bmatrix} = J_+ \oplus J_-$$

where $J_+ \in \mathcal{C}^{n_1 \times n_1}$ and $J_- \in \mathcal{C}^{n_2 \times n_2}$ are the Jordan blocks with $\mathrm{Re}(\sigma(A)) > 0$ and $\mathrm{Re}(\sigma(A)) < 0$, respectively, and $n = n_1 + n_2$. Applying the sign function to both sides of Equation (1), we obtain

$$\mathrm{sign}(A) = M\ \mathrm{sign}(J)\ M^{-1} \ .$$

The matrix sign of the Jordan blocks determine the sign of A as follows:

$$\mathrm{sign}(A) = M \begin{bmatrix} \mathrm{sign}(J_+) & 0 \\ 0 & \mathrm{sign}(J_-) \end{bmatrix} M^{-1} = M \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} M^{-1} \ . \tag{2}$$

Equation (2) shows that the Jordan blocks corresponding to the positive (negative) eigenvalues are mapped to positive (negative) identity matrices whose dimensions are the same as the number of positive (negative) eigenvalues. It follows that $S = \mathrm{sign}(A)$ is a diagonalizable matrix which commutes with A and is a square root of the identity, i.e.,

$$S^2 = I \ \text{and} \quad AS = SA \ . \tag{3}$$

Equation (3) is a quadratic equation in S, and can be solved by Newton's method [13]. Another definition, which is based on integral representation, is given in [13]. It uses the integral formula

$$A^+ = \frac{1}{2\pi j} \int_C (zI - A)^{-1} dz \ ,$$

where C is a simple closed contour in $\mathcal{C}^+$, containing the eigenvalues of A with positive real part. Using the equality $A^+ = (\mathrm{sign}(A) + I)/2$, we obtain an integral expression for $\mathrm{sign}(A)$ as follows:

$$\mathrm{sign}(A) = \frac{2A}{\pi} \int_0^\infty (y^2 I + A^2)^{-1} dy \ .$$

The parallel computation of the matrix sign function has recently received attention in order to deal with large matrices [4, 12]. In the following we give brief descriptions of the previously proposed parallel algorithms along with the new parallel algorithm. The analysis of these parallel algorithms are performed by counting the number of arithmetic operations and communication steps per iteration. The number of iterations needed for the convergence is a function of several factors, e.g., the size and condition of the matrix, properties of the algorithm, etc. The implementation results on an 8-processor Meiko CS-2 multiprocessor using the PVM software are summarized in Section 5.

2 The Parallel Newton Iteration

This algorithm is a slightly modified version of the symmetric pivoting algorithm of [4]. The original algorithm has been applied to the solution of the algebraic Riccati equation, where the iteration is carried on a matrix pencil with Hamiltonian-like structure. The structure of the pencil has the property that multiplying with an anti-diagonal identity matrix would convert it to a symmetric matrix. In this paper we have implemented a modified version of this algorithm for finding the matrix sign function of a general nonsymmetric matrix. The algorithm is based on parallel factorization of the iteration matrix $S[k] \in \mathcal{R}^{m \times m}$ at each step of the algorithm. We obtain the inverse of the symmetric matrix $S[k]$ in parallel, and compute its determinant d_k in order to calculate the scaling factor γ_k. The algorithm starts with $S[0] = A$, and proceeds using the iteration

$$S[k+1] = \frac{1}{2}(\frac{1}{\gamma_k} S[k] + \gamma_k S^{-1}[k]) \ . \tag{4}$$

The determinantal scaling factor [7] is given as

$$\gamma_k = |\det S[k]|^{1/m} \ .$$

The inverse of the iteration matrix $S[k]$ is computed, and the scaling factor γ_k is obtained during the parallel LU decomposition. The iteration matrix $S[k]$ is distributed among p processors in a column-wrapped fashion so that each processor works on an array of $n \times n/p$ elements. The parallel LU decomposition requires $O(n^3/p)$ arithmetic operations and $O(n)$ communication steps. The scaling and matrix addition requires $O(n^2/p)$ arithmetic steps. Finally, each processor sends its portion of the iteration matrix to the rest of the processors in order to calculate $S[k+1]$. This operation is called a multi-node broadcast operation. The details of the multi-node broadcast operation for various parallel architectures can be found in [2]. We denote the communication time of the multi-node broadcast operation of a single matrix element by B. Since n^2/p matrix elements are being broadcast, the update operation requires $O(n^2 B/p)$ communication steps. Thus, a single Newton iteration requires $O(n^3/p)$ arithmetic operations with a communication penalty of $O(n^2 B/p)$.

3 The Partial Fraction Expansion Algorithm

The most common method for finding sign(A) is Newton's method which is globally convergent for matrices with nonzero eigenvalues. This method can also be extended to globally convergent rational iterations of arbitrary order. Howland [5] derived a closed form formula for a desired degree rational iteration from the following error relation

$$\frac{s_{k+1} - 1}{s_{k+1} + 1} = \left(\frac{s_k - 1}{s_k + 1}\right)^p$$

from which we solve for s_{k+1} as

$$s_{k+1} = \frac{(s_k + 1)^p + (s_k - 1)^p}{(s_k + 1)^p - (s_k - 1)^p} \ .$$

This is the principal Padé iteration of order p [6]. Recently it is noted in [8] that this rational approximation can be represented as

$$s_{k+1} = \tanh(p \ \mathrm{arctanh} \ s_k) \ ,$$

where

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \ .$$

It is also shown in [8] that the above function for even p can be expanded in partial fractions as

$$\tanh(p \operatorname{arctanh}(s)) = \frac{s}{p}\sum_{i=0}^{p-1} \frac{1}{\sin^2(\frac{(2i+1)\pi}{4n}) + \cos^2(\frac{(2i+1)\pi}{4n})s^2} \tag{5}$$

Using the partial fraction expansion of (5), a parallel iteration for the matrix sign function is obtained [12]. The algorithm starts with $S[0] = A$, and uses the iteration formula

$$S[k+1] = S[k]\sum_{i=1}^{p} \frac{1}{p}(\alpha_i^2 I + \beta_i^2 S^2[k])^{-1} \ , \tag{6}$$

where p is the number of processors and

$$\alpha_i = \sin\left(\frac{(2i-1)\pi}{4n}\right) \quad \text{and} \quad \beta_i = \cos\left(\frac{(2i-1)\pi}{4n}\right) \ .$$

Each step of the parallel sign function iteration starts with parallel squaring of the iteration matrix $S[k]$, which requires $O(n^3/p)$ parallel arithmetic steps. After $S^2[k]$ is computed, it is distributed among the processors, and each processor sequentially computes $(\alpha_i^2 I + \beta_i^2 S^2[k])^{-1}$. This step requires $O(n^2/p)$ communication steps and $O(n^3)$ arithmetic operations. The summation of (6) is obtained using a binary tree, which requires $O(n^2 \log p)$ communication and arithmetic steps. After the sum is obtained, it is multiplied by $S[k]$ in parallel to obtain $S[k+1]$. This step requires $O(n^3/p)$ arithmetic operations. Thus, a single iteration step requires approximately $O(n^3/p)$ arithmetic operations and $O(n^2 \log p)$ communication steps.

4 The Parallel Continued Fraction Algorithm

The proposed algorithm is based on parallelization of the continued fraction algorithm [3]. This algorithm employs the inverse square-root of a matrix using the continued fraction expansion. The iterative algorithm for computing the inverse of the principal square root of the complex matrix $A \in C^{n\times n}$ is stated as follows:

$$\begin{aligned} \begin{bmatrix} P_1 \\ Q_1 \end{bmatrix} &= \begin{bmatrix} I \\ I \end{bmatrix} , \\ \begin{bmatrix} P_j \\ Q_j \end{bmatrix} &= \begin{bmatrix} I & I \\ A & I \end{bmatrix} \begin{bmatrix} P_{j-1} \\ Q_{j-1} \end{bmatrix} , \\ \lim_{j\to\infty} P_j Q_j^{-1} &= (\sqrt{A})^{-1} , \end{aligned} \tag{7}$$

where $P_j, Q_j \in C^{n\times n}$. By replacing the block element A with A^2 in the iteration matrix of (7), we obtain an iterative algorithm for computing the inverse square root of the square of a complex matrix, which in turn can be used for the computation of the matrix sign function due to the following definition of the matrix sign function:

$$\operatorname{sign}(A) = A(\sqrt{A^2})^{-1} = A^{-1}(\sqrt{A^2}) \ .$$

The continued fraction based matrix sign function algorithm starts with $S[0] = A$, and uses the iteration:

$$S[k+1] = S[k]\left(\sqrt{S^2[k]}\right)^{-1} \ .$$

For each value of k, we compute an approximation for the inverse of the principal square root of $S^2[k]$. This is achieved by iterating the continued fraction algorithm r times for $j = 1, 2, \ldots, r$. The matrix sign function algorithm starts with $S[0] = A$, and iteratively computes the sign function of A by going through a series of baby-steps and giant-steps, corresponding to the computation of an r-step approximation for the inverse of the principal square root and the computation of the new value of S, respectively. Thus, the iteration for the inverse square root is modified for computing the matrix sign function as follows:

$$\begin{array}{lrcl}
\textsf{Start:} & S[0] & = & A \; , \\
\textsf{Baby-Step:} & P_1[k] & = & I \; , \\
 & Q_1[k] & = & I \; , \\
j = 2, 3, \ldots, r: & P_j[k] & = & P_{j-1}[k] + Q_{j-1}[k] \; , \\
 & Q_j[k] & = & S^2[k] P_{j-1}[k] + Q_{j-1}[k] \; , \\
\textsf{Giant-Step:} & S[k+1] & = & S[k] P_r[k] Q_r^{-1}[k] \; .
\end{array} \tag{8}$$

A baby-step corresponds to the computation of $P_r[k]$ and $Q_r[k]$ using $S[k]$ and the initial values $P_1[k] = Q_1[k] = I$. There are no matrix inversions during this phase. The number of iterations during the baby-step phase is equal to r, which is predetermined. We start with $P_1[k] = I$ and $Q_1[k] = I$ and compute $P_j[k]$ and $Q_j[k]$ for $j = 2, 3, \ldots, r$, using the continued fraction based iterative algorithm for the inverse square root. A giant-step, on the other hand, corresponds to the computation of $S[k+1]$, using $S[k]$, $P_r[k]$, and $Q_r[k]$. There is a single matrix inversion during the giant-step. The number of giant-step iterations is a function of the convergence properties of the algorithm, the input matrix, as well as the baby-step length r.

- Distribute $P_{j-1}[k]$ and $Q_{j-1}[k]$ among processors in a column wrapped fashion.
- Compute $S^2[k]$ using parallel matrix multiply.
- Compute $S^2[k]P_{j-1}[k]$ using parallel matrix multiply.
- Add the corresponding columns of $P_{j-1}[k]$ and $Q_{j-1}[k]$ to obtain $P_j[k]$.
- Add $S^2[k]P_{j-1}[k]$ and $Q_{j-1}[k]$ in a similar fashion to obtain $Q_j[k]$.
- Solve in parallel for $S[k+1]$ in $Q_r[k]S[k+1] = S[k]P_r[k]$.

The proposed algorithm achieves data parallelism by distributing the matrices among the processors and obtaining the LU decomposition and matrix products in parallel. With $p < n$ processors, computing $S^2[k]$ and $S^2[k]P_{j-1}[k]$ and updating the new iterate on each processor requires $O(n^3/p)$ arithmetic steps and $O(n^2B/p)$ communication steps, where B is the communication overhead of a multi-node broadcast operation. Adding the corresponding rectangular arrays $P_{j-1}[k]$, $S^2[k]P_{j-1}[k]$ and $Q_{j-1}[k]$ to obtain $P_j[k]$ and $Q_j[k]$ requires n^2/p arithmetic steps and $O(n^2B/p)$ communication steps. Finally parallel solution of the linear system requires $O(n^3/p)$ parallel arithmetic operations. Therefore each giant-step requires approximately $O(rn^3/p)$ arithmetic steps and $O(rn^2B/p)$ communication steps where r is the baby-step length.

5 Implementation Results and Conclusions

We have implemented these three parallel matrix sign function algorithms on an 8-processor partition of a Meiko CS-2 multiprocessor, in which each node is a Sparc processor equipped with 256 MBytes of memory. The algorithms are implemented using the PVM software. In our experiments, we have computed the sign functions of matrices of dimensions ranging from 128 to 1024. The matrices are generated randomly with geometrically distributed eigenvalues. In Table 1, we give the parallel times for the three algorithms as a function of time for $p = 2, 4$, and 8 processors.

Table 1: The parallel times for the algorithms (in seconds).

	$p = 2$			$p = 4$			$p = 8$		
Size	N	PFE	CF	N	PFE	CF	N	PFE	CF
128	0.221	0.070	0.052	0.184	0.058	0.043	0.153	0.047	0.037
256	0.504	0.186	0.161	0.387	0.143	0.124	0.297	0.109	0.094
384	0.985	0.422	0.312	0.703	0.301	0.230	0.502	0.212	0.168
512	1.463	0.700	0.601	0.975	0.464	0.401	0.650	0.314	0.273
640	2.242	1.192	1.024	1.401	0.745	0.640	0.875	0.481	0.407
768	3.924	2.086	1.792	2.451	1.301	1.120	1.529	0.843	0.712
896	7.065	3.754	3.224	4.420	2.340	2.017	2.750	1.517	1.288
1024	12.993	6.941	5.967	8.146	4.325	3.729	5.100	2.807	2.379

The number of iteration steps for the the partial fraction algorithm is a function of the number of processors (the order of the summation) and the size of the matrix. For the continued fraction algorithm the number of the giant-steps depends on r. A thorough analysis of the dependency of the number of giant-steps on r is given in [3]. For this implementation we selected r to be 4. In Table 2, we tabulate the number of iterations for the algorithms as a function of the matrix size and the number processors.

Table 2: The number of iterations for the algorithms.

	$p = 2$			$p = 4$			$p = 8$		
Size	N	PFE	CF	N	PFE	CF	N	PFE	CF
128	13	9	8	13	6	8	13	5	8
256	13	9	7	13	6	7	13	5	7
384	13	9	7	13	6	7	13	5	7
512	14	9	7	14	6	7	14	5	7
640	14	8	7	14	6	7	14	4	7
768	14	8	7	14	5	7	14	4	7
896	14	8	7	14	5	7	14	4	7
1024	15	7	7	15	5	7	15	4	7

Our implementation results show that Newton's method is the slowest of all three algorithms, mainly because of the high number of iterations. For example, for matrix size 1024, Newton's method requires nearly twice the time required by the partial fraction expansion algorithm. Furthermore, comparing the other two algorithms to one another, we conclude that the continued fraction algorithm is slightly faster than the partial fraction expansion algorithm. Although the partial fraction algorithm requires the fewest number of iterations for $p = 4$ or 8, its total time is slightly larger than the continued fraction algorithm, This is mainly due to the fact that the partial fraction expansion algorithm requires the summation of n^3 matrix elements distributed over p processors, introducing a communication penalty of $O(n^3 \log p)$ at each step which considerably lengthens the total time.

References

[1] N. Akar and K. Sohraby. An invariant subspace approach in M/G/1 and G/M/1 type Markov chains. Technical report, Computer Science Telecommunications, University of Missouri, Kansas City, 1995.

[2] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation.* Englewood Cliffs, N.J: Prentice-Hall, 1989.

[3] Ç. K. Koç, B. Bakkaloğlu, and L. S. Shieh. Computation of the matrix sign function using continued fraction expansion. *IEEE Transactions on Automatic Control*, 39(8):1644–1647, 1994.

[4] J. D. Gardiner and A. J. Laub. Parallel algorithms for algebraic Riccati equations. *International Journal of Control*, 44(6):1317–1333, 1991.

[5] J. L. Howland. The sign matrix and the seperation of the matrix eigenvalues. *Linear Algebra and its Applications*, 49:221–232, 1983.

[6] C. Kenney and A. Laub. Rational iterative methods for the Matrix Sign Function. *SIAM Journal on Matrix Analysis and Applications*, 12(2):273–291, 1991.

[7] C. Kenney and A. J. Laub. On scaling Newton's method for polar decomposition and the matrix sign function. *SIAM Journal on Matrix Analysis and Applications*, 13(3):688–706, 1992.

[8] C. Kenney and A. J. Laub. A hyperbolic tangent identity and geometry of Padé sign function iterations. *Numerical Algorithms*, 7:111–128, 1994.

[9] C. Kenney and A. J. Laub. Polar decomposition and matrix sign function condition estimates. *SIAM Journal on Scientific and Statistical Computing*, 12(3):488–504, 1991.

[10] A. J. Laub. Invariant subspace methods for the numerical solution of Riccati equations. In S. Bittanti, A. J. Laub, and J. C. Willems, editors, *The Riccati Equation*, pages 163–196. New York, NY: Springer-Verlag, 1991.

[11] R. L. Mattheys. Stability analysis via the extended matrix sign function. *Proc. IEE*, 125(3):1057–1078, 1978.

[12] P. Pandey, C. Kenney, and A. J. Laub. A parallel algorithm for the matrix sign function. *International Journal of High-Speed Computing*, 2(2):181–191, 1990.

[13] J. D. Roberts. Linear model reduction and solution of the algebraic Riccati equation by use of the sign function. *International Journal of Control*, 32(4):677–687, 1980.

Parameterized Parallel Successive Over-Relaxation Algorithms for Linear Arrays

Gerard G.L. Meyer
E.C.E. Department
The Johns Hopkins University
Baltimore, MD 21218

Mike Pascale
Westinghouse Electric Corporation
Baltimore, MD 21203

Abstract

This paper presents a family of Successive Over-Relaxation (SOR) algorithms for Partial Differential Equations (PDEs) that contains Jacobi, red-black, and Gauss-Seidel and their relaxed versions. The family is parameterized by the sweeping order, the number of processors available P, the communication grain aggregation parameter U, and obtain a refined estimate $u^{(R)}$ of the solution of an elliptic PDE given an initial estimate $u^{(0)}$ by computing R iterations of SOR. When the number of processors P is small compared to $\max(M, N)$, where M and N describe the size of the PDE discretization grid, efficiencies in excess of 90% with the generally superior Gauss-Seidel ordering are obtained on a distributed memory array architecture.

Key words: Parallel elliptic PDE solution, Gauss-Seidel SOR, Red-Black SOR, Jacobi SOR, processor and memory architectures.

1 Introduction

In this paper, we focus our attention on 2-D second order linear PDEs of the form

$$\begin{aligned} a(x,y)\frac{\partial^2 u}{\partial x^2} + b(x,y)\frac{\partial u}{\partial x} + c(x,y)\frac{\partial^2 u}{\partial y^2} & \\ +d(x,y)\frac{\partial u}{\partial y} + e(x,y)\frac{\partial^2 u}{\partial x \partial y} + f(x,y)u &= g(x,y) \end{aligned} \tag{1}$$

and their numerical solution via Successive Over-Relaxation (SOR) methods. Given an initial estimate $u^{(0)}$, the SOR methods [7] obtain a refined estimate $u^{(R)}$ of the solution of Eq. (1) discretized over an M by N grid by using R iterations which iteratively improve each of the discretized solution estimate components $u_{m,n}^{(r)}$ by combining the previous estimate $u_{m,n}^{(r-1)}$ with recent estimates of its north, west, east, and south neighbors. Thus

$$\begin{aligned} u_{m,n}^{(r)} = u_{m,n}^{(r-1)} - \omega^{(r)} \Big(& \beta_{m,n,N} u_{m-1,n}^{(*N)} + \beta_{m,n,W} u_{m,n-1}^{(*W)} + u_{m,n}^{(r-1)} \\ & + \beta_{m,n,E} u_{m,n+1}^{(*E)} + \beta_{m,n,S} u_{m+1,n}^{(*S)} - \gamma_{m,n} \Big) \end{aligned} \tag{2}$$

for $r = 1, 2, \ldots, R$ and for all $(m, n) \in \Omega^\circ$, given the relaxation sequence $\omega^{(r)}$ for $r = 1, 2, \ldots, R$, an initial discretized solution estimate $u_{m,n}^{(0)}$ for all $(m, n) \in \Omega^\circ$ and boundary conditions $u_{m,n}^{(R)} = u_{m,n}^{(0)}$ for all $(m, n) \in \partial\Omega$ where

$$\Omega = \left\{ (m,n) \,\middle|\, \begin{array}{l} m \in \{0, 1, \ldots, M+1\} \\ n \in \{0, 1, \ldots, N+1\} \end{array} \right\},$$

and where each sweeping ordering parameter $*_N$, $*_W$, $*_E$, and $*_S$ takes a value of r or $r-1$ and implies a sequence of precedence among the computations of $u_{m,n}^{(r)}$. We obtain our family of parallel SOR algorithms by segmenting the SOR algorithms into arithmetic grains, parameterizing the assignment of the arithmetic grains to at most P parallel processes intended for execution on P processors, and parameterizing the number of arithmetic grains computed between communications events. To evaluate the complexity and performance of the parallel algorithms presented here, we assume that $\omega^{(r)}$ and R are known and that the discretization grid is static.

As the numerical performance and parallelism of a given algorithm depends on the ordering parameters [1, 2, 3, 6], we will consider Jacobi (J), red-black Gauss-Seidel (RB), and natural Gauss-Seidel (GS) orderings. In the Jacobi ordering, $*_N = *_W = *_E = *_S = r-1$. Thus with the J ordering, all components at iteration r may be computed in parallel. In the RB ordering, the components of u are divided into two groups; $u_{m,n}$ is red if $m+n$ is even, and black if $m+n$ is odd. Red components at iteration r are updated using black components from iteration $r-1$, that is, $*_N = *_W = *_E = *_S = r-1$. Black components at iteration r are updated using red components from iteration r, that is, $*_N = *_W = *_E = *_S = r$. Thus with the RB ordering, all red components may be computed in parallel followed by the computation of all black components in parallel. In the GS ordering, $*_N = *_W = r$, and $*_E = *_S = r-1$, and thus all components with identical values of $m+n$ may be computed in parallel. These ordering are summarized in Table 1.

sweeping order		$*_N$	$*_W$	$*_E$	$*_S$
Jacobi (J)		$r-1$	$r-1$	$r-1$	$r-1$
red-black (RB)	red	$r-1$	$r-1$	$r-1$	$r-1$
	black	r	r	r	r
Gauss-Seidel (GS)		r	r	$r-1$	$r-1$

Table 1: Ordering parameters

The organization of the paper is as follows. The architecture and architectural parameters are presented in Section 2. In Section 3, we develop a parameterized family of SOR algorithms by segmenting the SOR into arithmetic grains, by parameterizing the assignment of the arithmetic grains to parallel processes for a uniform work distribution, and by parameterizing the communication granularity. Latency as a function of design parameters is characterized in Section 4 and exemplified in Section 5. Section 6 summarizes our conclusions.

2 Architecture and Architectural Parameters

The target architecture and associated software protocol consists of P processors connected in a linear array with bi-directional communication links. Each processor requires time τ_a to execute an arithmetic instruction, where τ_a includes the cost of instruction fetch and decode, operand fetch and save, caching, operand index calculation, loop overhead, etc.. A communication link requires time $\tau_c(w) = \tau_s + w\tau_w$ to transfer a w-word message across a link where τ_s is the message start-up time and τ_w is the per word transfer time. We also assume that the software protocol requires the computational nodes to pause until communications requests have been satisfied, that is, the arithmetic computations which are dependent upon an input message are blocked until the last word of that message has been received [5]. Furthermore, output messages are not initiated until after the last word of a message has been computed.

The capabilities of the P processors will classify the architecture as either *non-concurrent* or *concurrent.* In the non-concurrent case, the P processors can perform either uni-directional communications across one communication link or execute arithmetic instructions at any given time. In the concurrent case, the processors are capable of performing uni-directional communications on all communication links simultaneously with the execution of arithmetic instructions.

3 The Parameterized Family of SOR Algorithms

We obtain the family of parallel SOR algorithms by segmenting the SOR algorithm into arithmetic grains, parameterizing the assignment of the arithmetic grains to at most P parallel processes intended for execution on P processors, and parameterizing the number of arithmetic grains computed between communications events.

We define an *arithmetic grain*, denoted by its output $u^{(r)}_{m,n}$, to consist of the operations of Eq. (2) for fixed m, n, and r. The arithmetic complexity and communication among grains are summarized in Table 2. Thus R iterations of SOR consist of MNR arithmetic grains whose execution require $11MNR$ operations.

operations			input		output	
+	×	total	variables	words	variables	words
6	5	11	$u^{(*N)}_{m-1,n}, u^{(*W)}_{m,n-1}, u^{(r-1)}_{m,n}, u^{(*E)}_{m,n+1}, u^{(*S)}_{m+1,n}$	5	$u^{(r)}_{m,n}$	1

Table 2: Arithmetic Grain Computation and Communication Complexities

The assignment of the MNR grains to P processes is dictated by P *arithmetic grain aggregation coefficients* $h_1, h_2, \ldots, h_P$, where

$$\lfloor N/P \rfloor \leq h_p \leq \lceil N/P \rceil, p = 1, 2, \ldots, P, \text{ and } \Sigma_{p=1}^{P} h_p = N.$$

The arithmetic grain aggregation coefficients in turn allow us to define the *cumulative arithmetic grain aggregation coefficients* $H_0, H_1, \ldots, H_P$ where

$$H_0 = 0, H_p = H_{p-1} + h_p, p = 1, 2, \ldots, P.$$

The arithmetic grains assigned to process p for $p = 1, 2, \ldots, P$ are $u^{(r)}_{m,n}$ for all $m = 1, 2, \ldots, M$, for all $n = H_{p-1} + 1, H_{p-1} + 2, \ldots, H_p$, and for all $r = 1, 2, \ldots, R$. The relationship between the discretization grid and the processing array is shown in Figure 1.

As the number of arithmetic grains assigned to process p is $h_p MR$, the number of arithmetic operations executed by process p is $11 h_p MR$. For each $r = 1, 2, \ldots, R$, each process p is dependent upon receiving a west boundary of M words consisting of $u^{(*W)}_{m,H_{p-1}}$ for $m = 1, 2, \ldots, M$ and an east boundary of M words consisting of $u^{(*E)}_{m,H_p+1}$ for $m = 1, 2, \ldots, M$. In addition, each process p for each $r = 1, 2, \ldots, R$ must send

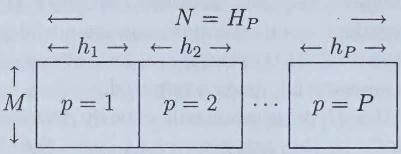


Figure 1: Discretization Grid and Processing Array Relationship

a west boundary of M words consisting of $u^{(r)}_{m,H_{p-1}+1}$ for $m = 1, 2, \ldots, M$ and an east boundary of M words consisting of $u^{(r)}_{m,H_p}$ for $m = 1, 2, \ldots, M$. The total arithmetic and communication complexities for process p are summarized in Table 3.

arithmetic		input to process p (words)		output from process p (words)	
grains	operations	$p-1$	$p+1$	$p-1$	$p+1$
h_pMR	$11h_pMR$	MR	MR	MR	MR

Table 3: Grain Computation and Communication Complexities for Process p

We now choose the order in which arithmetic grains are executed by each process taking into account their interprocess dependencies. Since the GS sweeping dependencies are a superset of the RB dependencies which are a superset of the J sweeping dependencies, we will develop an arithmetic grain ordering for the GS sweeping which will inherently satisfy the dependencies for the other two sweepings. In the case of the RB sweeping, we relabel the indices so that all red arithmetic grains precede black arithmetic grains. The execution of arithmetic grain $u^{(r)}_{m,n}$ is dependent upon the variables as given in Table 2. To satisfy these dependencies, we choose the arithmetic grains as illustrated in Figure 2 to be executed from top to bottom among rows and from left to right within a row.

A *communication grain* is the communication of a single word of boundary information by any process p to the west process $p-1$ or to the east process $p-1$. There is an input and output communication grain associated with each arithmetic grain on the left and right edges of Figure 2. We choose the order in which the communication grains are executed to be the order in which the corresponding boundary information is needed and generated by each process according to the arithmetic grain execution ordering described above. The order communication grains between process p and western process $p-1$ are executed is from top to bottom. Likewise, the order in which communication grains between process p and eastern process $p+1$ are executed is from top to bottom.

Let an *arithmetic step* be the contiguous arithmetic grains executed between communications events and let U be the number of communication grains in any message. The choice of U induces the number of arithmetic grains in each arithmetic step. Because the time to communicate a w-word message is $\tau_c(w) = \tau_s + w\tau_w$, longer messages, that is large U, result in a smaller average per word transfer time; reducing overall latency. However, longer messages can also contribute to delaying computations on processors which are dependent upon message data; increasing overall latency. Thus expressing latency as a function U affords a means to determine this trade-off optimally.

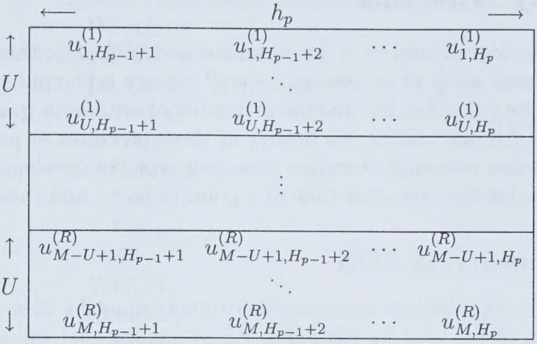


Figure 2: Arithmetic Steps for Process p

The number of input and output words to each process is MR and therefore, the number of messages to and from each process is

$$S = \left\lceil \frac{MR}{U} \right\rceil .$$

The reception by process p of each U-word messages from process $p-1$ with the corresponding message from process $p+1$ enables the computation of Uh_p arithmetic grains which generate a pair of U word messages; one needed by process $p-1$ and the other needed by process $p+1$. Thus the execution of Uh_p arithmetic grains is bracketed by communications events which defines an arithmetic step and therefore the number of arithmetic steps is S.

We now define five subroutines that are common to both concurrent and non-concurrent parallel implementations of the SOR algorithm. Each subroutine call of `Comp(s)`, `Wout(s)`, `Eout(s)`, `Win(s)`, and `Ein(s)` executes approximately $1/S$ of the total of arithmetic grains, west output grains, east output grains, west input grains, or east input grains, respectively where the argument $s \in [1, 2, \ldots, S]$ specifies which $1/S$ of the total grains are executed for a particular subroutine call. For instance, if we let $u_{i,j} = u^{(r)}_{m,n}$ with $i = M(r-1)+m$ and $j = n$, then `Comp(s)` executes $u_{i,j}$ for $i = U(s-1)+1, U(s-1)+2, \ldots, Us$ and $j = H_{p-1}+1, H_{p-1}+2, \ldots, H_p$.

Also, the parallel execution of instructions 1, 2, ..., n will be indicated by

instruction 1 // instruction 2 // ... // instruction n .

At the conclusion of the execution of the algorithm, each process p will have computed S arithmetic steps for $s = 1, 2, \ldots, S$ totaling $g_p MR$ grains, sent S messages for $s = 1, 2, \ldots, S$ totaling MR words to process $p-1$, received S messages for $s = 1, 2, \ldots, S$ messages totaling MR words from process $p-1$, sent S messages for $s = 1, 2, \ldots, S$ totaling MR words to process $p+1$, and received S messages for $s = 1, 2, \ldots, S$ totaling MR words from process $p+1$ excluding the boundary processes $p = 1$ and $p = P$, where the communication to process $p-1$ and $p+1$ respectively, is null. The relative order of arithmetic step execution and message execution is discussed next.

4 Latency Analysis

In this section, the overall latency of the parameterized SOR algorithms are quantified for execution on a linear array of processors. Overall latency is computed by evaluating the latency of a process q that has the maximum number of arithmetic grains associated with it, that is, $h_q = \left\lceil \frac{N}{P} \right\rceil$, and adding the latency of those processes or portions of processes required prior to and following execution process q. For the development which follows, we denote the arithmetic execution time of a grain to be τ_g, and thus $\tau_g = 11\tau_a$.

4.1 Non-concurrent SOR

When arithmetic computations and communications cannot be done simultaneously, we use the algorithm of Figure 3 for the J and RB sweepings, and we use the algorithm of Figure 4 for the GS sweeping. To satisfy dependency constraints, we require $U \leq M$ in the J case, and $U \leq M/2$ in the RB and GS cases.

```
DO IN PARALLEL FOR $p = 1, \ldots, P$
  for $s = 1, 2, \ldots, S$
    Wout($s$)
    Eout($s$)
    Win($s$)
    Ein($s$)
    Comp($s$)
  end
END
$p$ = EVEN
```

```
DO IN PARALLEL FOR $p = 1, \ldots, P$
  for $s = 1, 2, \ldots, S$
    Ein($s$)
    Win($s$)
    Eout($s$)
    Wout($s$)
    Comp($s$)
  end
END
$p$ = ODD
```

Figure 3: Non-concurrent 1-D Jacobi/Red-Black Algorithm

In the J and RB cases, dependencies allow execution of the worst case process to begin immediately, and will proceed without blocking since its west and east neighbors have at most the same number of grains to compute at each arithmetic step. Thus the latency in the J and RB cases is bounded from above by L_{Jn} and L_{RBn} with

$$\begin{aligned} L_{Jn} = L_{RBn} &= S(4(\tau_s + U\tau_w) + h_q U \tau_g) \\ &= \left\lceil \frac{MR}{U} \right\rceil (4\tau_s + 4U\tau_w + \lceil N/P \rceil U \tau_g) . \end{aligned}$$

Minimizing over the communication granularity parameter U in the J case gives $U = M$ and

$$L_{Jn} = 4R\tau_s + 4MR\tau_w + MR\lceil N/P \rceil \tau_g,$$

and in the RB case gives $U = M/2$ and

$$L_{RBn} = 8R\tau_s + 4MR\tau_w + MR\lceil N/P \rceil \tau_g.$$

In the non-concurrent GS sweeping, dependencies block the execution of the worst case process q until processes $p = 1, 2, \ldots, q-1$ have executed their respective first triplet of input communications, arithmetic step, and output communications. Then execution

```
DO IN PARALLEL FOR $p = 1, \ldots, P$
    Wout(1)
    Ein(1)
    for $s = 1, 2, \ldots, S-1$
        Win($s$)
        Wout($s+1$)
        Comp($s$)
        Eout($s$)
        Ein($s+1$)
    end
    Win($S$)
    Comp($S$)
    Eout($S$)
END
```

Figure 4: Non-concurrent 1-D Gauss-Seidel Algorithm

of the worst case process proceeds unblocked since its west and east neighbors have at most the same number of grains to compute at each arithmetic step. When the worst case process concludes, processes $p = q+1, q+2, \ldots, P$ must execute their final triplet of input communications, arithmetic step, and output communications. The latency incurred before the process q loop can begin executing is

$$(1 + 2(q-2))(\tau_s + U\tau_w) + \sum_{p=1}^{q-1} h_p U \tau_g,$$

the latency incurred executing the process q loop is

$$4(S-1)(\tau_s + U\tau_w) + (S-1) h_q U \tau_g,$$

and the latency incurred following the process q loop is

$$2(p-q)(\tau_s + U\tau_w) + \sum_{p=q}^{P} h_p U \tau_g.$$

Thus the latency in the GS case is bounded from above by L_{GSn} with

$$L_{GSn} \;=\; \left(4\left\lceil\frac{MR}{U}\right\rceil + 2P - 5\right)(\tau_s + U\tau_w) + \left(\left(\left\lceil\frac{MR}{U}\right\rceil - 1\right)\left\lceil\frac{N}{P}\right\rceil U + NU\right)\tau_g.$$

4.2 Concurrent SOR

When arithmetic computations and communications can be done simultaneously, we use the algorithm of Figure 5 for the J and RB sweepings, and we use the algorithm of Figure 6 for the GS sweeping. To satisfy dependency constraints and to permit the concurrent execution of arithmetic grains and communication grains, we require $U \leq M/2$ in the J case and $U \leq M/4$ in the RB and GS cases.

As in the non-concurrent situation, in both the J and RB cases, dependencies allow execution of the worst case process to begin immediately, and to proceed without blocking

```
DO IN PARALLEL FOR $p = 1, \ldots, P$
    Wout(1) // Eout(1) // Win(1) // Ein(1)
    for $s = 1, 2, \ldots, S-1$
        Wout($s+1$) // Eout($s+1$) // Win($s+1$) // Ein($s+1$) // Comp($s$)
    end
    Comp($S$)
END
```

Figure 5: Concurrent 1-D Jacobi/Red-Black Algorithm

since its west and east neighbors have at most the same number of grains to compute at each arithmetic step. Thus the latency in the J and RB cases is bounded from above by L_{Jc} and L_{RBc} with

$$\begin{aligned} L_{Jc} = L_{RBc} &= (\tau_s + U\tau_w) + (S-1)\max\{\tau_s + U\tau_w, h_q U\tau_g\} + h_q U\tau_g \\ &= (\tau_s + U\tau_w) + \left(\left\lceil \frac{MR}{U} \right\rceil - 1\right)\max\left\{\tau_s + U\tau_w, \left\lceil \frac{N}{P} \right\rceil U\tau_g\right\} + \left\lceil \frac{N}{P} \right\rceil U\tau_g. \end{aligned}$$

If $\tau_s + U\tau_w \le \left\lceil \frac{N}{P} \right\rceil U\tau_g$, then

$$L_{Jc} = L_{RBc} = (\tau_s + U\tau_w) + \left\lceil \frac{MR}{U} \right\rceil U \left\lceil \frac{N}{P} \right\rceil \tau_g \approx (\tau_s + U\tau_w) + MR\left\lceil \frac{N}{P} \right\rceil \tau_g.$$

In the concurrent GS sweeping, dependencies block the execution of the worst case process q until processes $p = 1, 2, \ldots, q-1$ have executed their respective first triplet of input communications, arithmetic step, and output communications. Then execution

```
DO IN PARALLEL FOR $p = 1, \ldots, P$
    Wout(1)
    Wout(2) // Ein(1)
    Wout(3) // Win(1)
    Wout(4) // Win(2) // Ein(2)
    for $s = 2, 3, \ldots, S-3$
        Wout($s+3$) // Win($s+1$) // Ein($s+1$) // Comp($s$) // Eout($s-1$)
    end
    Win($S-1$) // Ein($S-1$) // Comp($S-2$) // Eout($S-3$)
    Win($S$) // Ein($S$) // Comp($S-1$) // Eout($S-2$)
    Comp($S$) // Eout($S-1$)
    Eout($S$)
END
```

Figure 6: Concurrent 1-D Gauss-Seidel Algorithm

of the worst case process proceeds unblocked since its west and east neighbors have at most the same number of grains to compute at each arithmetic step. When the worst case process concludes, processes $p = q+1, q+2, \ldots, P$ must execute their final triplet of input communications, arithmetic step, and output communications. The latency incurred before process q can begin executing is

$$\sum_{p=1}^{q-1} (\tau_s + U\tau_w + \max\{\tau_s + U\tau_w, h_p U\tau_g\}),$$

the latency incurred executing process q is

$$(\tau_s + U\tau_w) + S\max\{\tau_s + U\tau_w, h_q U\tau_g\},$$

and the latency incurred following process q is

$$\sum_{p=q+1}^{P} (\tau_s + U\tau_w + \max\{\tau_s + U\tau_w, h_p U\tau_g\}).$$

Thus the latency in the GS case is bounded from above by L_{GSc} with

$$\begin{aligned} L_{GSc} = &\sum_{p=1}^{P} (\tau_s + U\tau_w + \max\{\tau_s + U\tau_w, h_p U\tau_g\}) \\ &+(S-1)\max\{\tau_s + U\tau_w, \lceil N/P \rceil U\tau_g\}. \end{aligned}$$

If $\tau_s + U\tau_w \leq h_p U\tau_g$ for all p, then

$$L_{GSc} = P(\tau_s + U\tau_w) + \left(\left(\left\lceil \frac{MR}{U} \right\rceil - 1\right) \left\lceil \frac{N}{P} \right\rceil U + NU\right) \tau_g.$$

Once again, given architectural parameters P, τ_g, τ_w and τ_s, and problem parameters M, N, and R, one can plot L_{GSc} as a function of the communication granularity U, and obtain the value of U which minimizes latency. Although it has been customary to use either $U = 1$ or $U = M$, we see in the concurrent latency example given by the solid line in Figure 7 that the optimal value of U_c is 3 and the resulting latency is less than half that obtained by using the customary values of $U = 1$ or $U = N$.

5 Example

In this section, we illustrate the benefit of parameterizing the communication granularity U in the GS case. Given architectural parameters P, τ_g, τ_w and τ_s, and problem parameters M, N, and R, one can plot L_{GSn} or L_{GSc} as a function of the communication granularity U, and obtain the values of U which minimize latency. Figure 7 illustrates L_{GSn} (dashed line) and L_{GSc} (solid line) for the nominal architectural parameters $P = 100$, $\tau_a = 1\mu$s, $\tau_w = 1\mu$s, and $\tau_s = 10\mu$s, and for the nominal problem parameters $M = N = 500$ and $R = 10$. The multiprocessor latency bound in this example is $11MNR\tau_a/P = 27.5$ms. We see that in the non-concurrent architecture case, the optimal value $U = 16$ yields the minimal latency $L_{GSn} = 74$ms and a corresponding efficiency of 37% which is less than half the latency obtained by using the customary values of $U = 1$ or $U = N$. Likewise, we see that in the concurrent architecture case, the optimal value $U = 3$ yields the minimal latency $L_{GSn} = 30$ms and a corresponding efficiency of 90% which is again less than half the latency obtained by using the customary values of $U = 1$ or $U = N$.

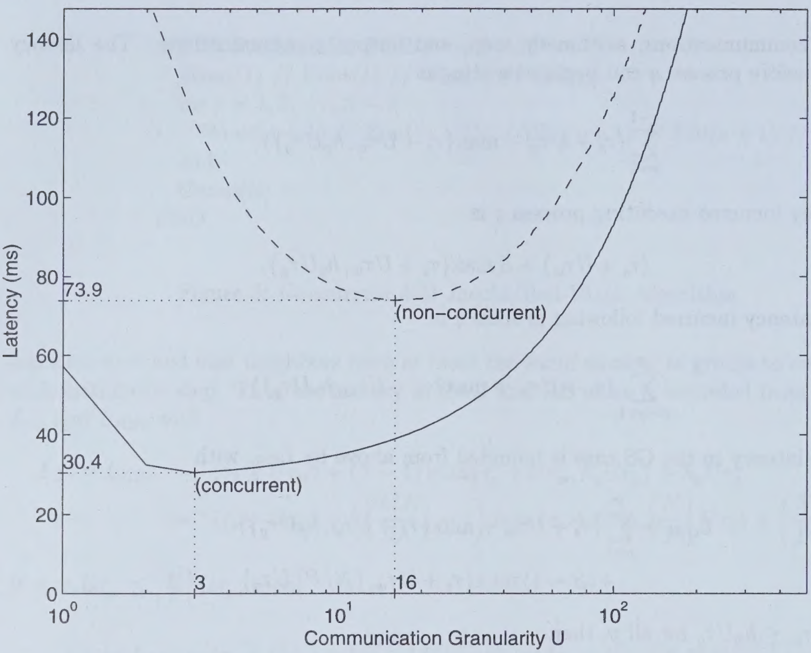


Figure 7: Gauss-Seidel Latency vs. Communication Granularity
$P = 100$, $\tau_a = .1\mu\text{s}$, $\tau_w = 1\mu\text{s}$, $\tau_s = 10\mu\text{s}$, $M = N = 500$, $R = 10$

6 Conclusions

The frequency in which intermediate results are communicated in a parallel processing environment has a significant impact on latency performance. Because the time to communicate a w-word message is $\tau_c(w) = \tau_s + w\tau_w$, longer messages result in a smaller average per word transfer time; reducing overall latency. However, longer messages can also delay computations on processors which are dependent upon message data; increasing latency. Expressing latency as a function of *communication granularity*, which is related to message length, allows optimal determination of this trade-off.

In particular, this paper presents a family of parallel algorithms for the solution of elliptic PDEs by SOR which is parameterized as a function of sweeping order; J, RB, or GS, and communication granularity U - the number of arithmetic operations computed within a process between communications which determines message length. Thus given a problem, one can determine or estimate the number of iterations required to achieve a desired accuracy improvement for each sweeping order R_J, R_{RB} and R_{GS}, find the optimal U and the corresponding latency for each case, and then choose the best SOR algorithm.

The algorithms presented here are intended for execution on a linear array which is parameterized by the number of processors P. Linear arrays were chosen because they are comparatively easy to design parallel algorithms for and are easy to program. From a hardware point of view, linear arrays are physically smaller, consume less power, are scalable, and comparatively inexpensive. Considering performance, if the efficiency of a parallel algorithm for a particular problem and linear array is high then little improvement can be expected in considering other topologies. For instance a nominal problem with $M = N = 500$ and $R = 10$ and a nominal concurrent architecture with $P = 100$, $\tau_a = .1\mu\text{s}$, $\tau_w = 1\mu\text{s}$, $\tau_s = 10\mu\text{s}$, gives a latency of 30.4ms compared to an ideal latency of 27.5ms which indicates an efficiency of 90%. Thus the maximum improvement which may be achieved by considering architectures other than linear arrays is 2.9ms.

We also make the remark that the family of SOR algorithms presented here distributes both the work and data of all R iterations evenly among processors. It is not always the case that work complexity corresponds with data complexity. In such cases it is generally advantageous to parameterize the distribution of work and express the arithmetic plus communication latency as a function of the parameterization to obtain the best parallel implementation as opposed to the customary approach of distributing data evenly. A case in point is the application of this parallel algorithm design approach to the QR factorization in [4].

References

[1] L.M. Adams and H.F. Jordan, Is SOR color-blind?, *J. Sci. Stat. Comput.* 7 (1986) 490-506

[2] C.-C.J. Kuo, T.F. Chan, and C. Tong, Two color Fourier analysis of iterative algorithms for elliptic problems with red/black ordering, *SIAM J. Sci Stat. Comput.* 11 (1990) 767-794

[3] C.-C.J. Kuo, B.C. Levy, Discretization and solution of elliptic PDEs - A digital signal processing approach, *Proc. IEEE* 12 (1990) 1808-1842

[4] G.G.L. Meyer and M. Pascale A family of Parallel QR Factorization Algorithms, *High Performance Computing Symposium '95* (1995) 95-106

[5] D.P. O'Leary and P. Whitman, Parallel QR factorization by Householder and modified Gram-Schmidt algorithms, *Parallel Computing* 16 (1990) 99-112

[6] J.M. Ortega and R.G. Voigt, Solution of partial differential equations on vector and parallel computers, *SIAM Review* 27 (1985) 149-240

[7] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in C, Cambridge University Press (1992) Chapter 19

Checking regularity of interval matrices by signed digraphs

Atsushi Nogi and Yoshiteru Ishida
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-01, Japan
atusi-n@is.aist-nara.ac.jp, ishida@is.aist-nara.ac.jp

Abstract This paper presents a new method for checking regularity of interval matrices using sign structure of the matrices. It is demonstrated that the information of sign structure of the matrices can reduce the computations for interval matrices. Further, it is shown that regularity of the original interval matrix can be checked by some vertex matrices.

1 Introduction

Precise analysis on large-scale and complex systems such as economic systems [1] and ecosystems [2] is often difficult due to the fact that numerical information of parameters of such systems is not always available. In these cases, qualitative and structural analysis with graphs plays an important role.

There are several different levels of abstraction such as sign systems or interval systems. However those systems have been studied rather independently and separately. Although graphical analysis has been extensively studied for sign matrices [3] [4], it has not yet been fully performed on interval matrices.

An interval matrix A^I is said to be regular if each matrix $A \in A^I$ is nonsingular. The problem of regularity of interval matrices is studied in [5], where a number of necessary and sufficient regularity conditions are given. However, 2^{n-1} linear systems must be solved to check those conditions.

In this paper we study regularity of interval matrices. We use qualitative structure of the matrix expressed by signed digraph. In the next section we provide the necessary notations and auxiliary results. In section 3 we show that regularity of interval matrices can be characterized by graph theoretical approach. It is demonstrated that the information of sign structure of the matrices can reduce the computations for interval matrices. Further, it is shown that regularity of the original interval matrix can be checked by more restrictive interval matrix obtained by fixing some elements at terminals of the original intervals. We develop an algorithm which verifies the regularity of interval matrices using the above result. We demonstrate the applicability of our algorithm by means of some experiments in section 4.

2 Notations and auxiliary results.

2.1 Interval and sign matrices

For two matrices $A, B \in R^{n\times n}$, $A \leq B$ denotes element-wise inequality. An interval matrix is defined as

$$A^I = \{A \in R^{n\times n} : \underline{A} \leq A \leq \overline{A}\},$$

where $\underline{A} = (\underline{a}_{ij})$ and $\overline{A} = (\overline{a}_{ij})$ are real $n \times n$ matrices. We also use the center matrix $A_c = \frac{1}{2}(\underline{A} + \overline{A})$ and the radius $\Delta = \frac{1}{2}(\overline{A} - \underline{A})$.

A matrix $A \in R^{n\times n}$ is said to be reducible if there is a permutation matrix P satisfying

$$PAP^T = \begin{pmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{pmatrix},$$

where B_{11} and B_{22} are square matrices; A is irreducible if it is not reducible. Reducibility of an interval matrix can be defined in a similar manner assuming that each non-zero element does not contain zero in its interval. We assume this throughout this paper.

An interval matrix can be permuted so that all its diagonal elements are negative if the matrix is irreducible and regular. We call this the standard form of a matrix. Such transformation of a matrix preserves its regularity.

Sign matrix is a matrix whose elements are one of $+, -$ or 0. Sign regularity is defined in the same way as that of interval matrices.

2.2 Signed digraphs and determinants

With a matrix $A \in R^{n\times n}$ we associate the signed digraph $D(A) = (V, E)$, having a set of vertices $V = \{1, 2, \cdots, n\}$ and a set of arcs E containing the arc (i, j) if and only if $a_{ij} \neq 0$. Each arc is associated with $+$ or $-$ sign. We call a sequence $(i_1, i_2, \cdots, i_r, i_1)$, where $i_1, i_2, \cdots, i_r$ are distinct vertices of $D(A)$ and each of the arcs $(i_1, i_2), \cdots, (i_r, i_1)$ belongs to E, a cycle of $D(A)$. A sign of a cycle is the product of signs of all arcs in the cycle.

By definition, a determinant for a matrix $A \in R^{n\times n}$ then can be formulated by:

$$\det A = \sum_{\phi} (sgn\ \phi) \prod_{i=1}^{n} a_{i,\phi(i)}, \tag{1}$$

where ϕ is an arbitrary permutation of integer $1, \cdots, n$ and $sgn\ \phi$ is the sign of the permutation ϕ. Any permutation ϕ can be uniquely factored into a product of disjoint permutation cycles which correspond to disjoint cycles of $D(A)$. We call the set of cycles a factor of $D(A)$. Let f_k $(k = 1, 2, \cdots, q)$ denote a factor, where k is the arbitrarily assigned index, and q is the number of nonzero terms of $\det A$. A sign of a factor is defined as the product of the signs of all the cycles in the factor and $(-1)^{m_k}$, where m_k is the number of cycles contained in the kth factor.

If a_{ij} of A is included in both positive and negative term of determinant expansion of A, it is called a sign-conflict element, and the corresponding arc (ij) of $D(A)$ is called a sign-conflict arc.

2.3 Sufficient condition for regularity of interval matrices

Thus far, most of necessary and sufficient conditions for regularity of interval matrices are difficult to apply. However there are some sufficient conditions that are easy to apply, e.g.:

Lemma 2.1 (Beeck 1975 [6]) *Let A^I be an interval matrix and A_c, Δ be the center matrix and radius of A^I respectively. Then if the nonnegative matrix $D = |A_c^{-1}|\Delta$ satisfies $\rho(D) < 1$, where $\rho(D)$ denotes the spectral radius of D, then A^I is regular.* □

We will argue the applicability of this condition in section 4.

3 The main result

In this section, we present the main result: computation for regularity of an interval matrix can be reduced by constructing more restrictive interval matrix based on its graph-theoretical property.

It is known that a sign matrix A^S is regular if and only if each of the terms in the expansion of the determinant of A^S are of the same sign. Qualitative results for sign matrices has been provided by Maybee[3] based on the cyclic analysis of determinants. The following is a graphical characterization for a sign matrix to be sign regular.

Lemma 3.1 (Maybee 1981 [3]) *A sign matrix A^S is regular if and only if each cycle of $D(A^S)$ has a negative sign in the standard form of A^S.* □

Example 1:

This example shows a case in which Lemma 3.1 can be directly applied to an interval matrix. Figure 1 shows the corresponding sign matrix and the digraph of A^I. Because $D(A)$ does not have any positive cycles, the sign matrix is regular, which in turn implies the regularity of A^I.

$$A^I = \begin{pmatrix} [-2,-1] & [-4,-2] & 0 \\ [\ 1,\ \ 3] & [-2,-1] & [-3,-1] \\ [-5,-4] & [\ 1,\ \ 2] & [-3,-1] \end{pmatrix}$$

□

$$\begin{pmatrix} - & - & 0 \\ + & - & - \\ - & + & - \end{pmatrix}$$

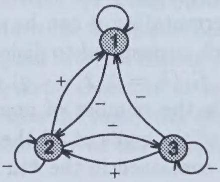


Figure 1: The sign matrix and its signed digraph $D(A)$

A Matrix in which each element takes terminal value of its interval is called a vertex matrix. There are 2^{n^2} vertex matrices for the interval matrix of $n \times n$ where all the intervals $[a,\ b]$ of elements are specified two distinct terminal values; $a \neq b$.

As the problem of checking regularity of interval matrices is known to be NP-hard[7], in general, combinatorial computation over vertex matrices need to be performed. Although we need to deal with not all 2^{n^2} vertex matrices but 2^{n-1} vertex matrices of a special form [5], it is still exponential.

Therefore we try to reduce the number of vertex matrices to be tested by fixing some intervals to either of their terminal values. Such fixing will be possible when the element of the interval matrix appears only in the terms of the same sign, i.e., it is not a sign-conflict element.

Whether or not the element is a sign conflict type can be known by the signed digraph without examining signs of all the terms in cofactor of the element. The next result characterizes a sign-conflict element by graph theoretical conditions based on the signs of cycles in the digraph.

Theorem 3.1 *Let A^I be an interval matrix. a_{ij} of A^I is a sign-conflict element if and only if either or both of the following conditions is satisfied.*

i) The arc (ij) in $D(A^I)$ is contained in both negative and positive cycles at the same time.

ii) The arc (ij) in $D(A^I)$ is contained in a cycle disjoint to a positive cycle. □

First observe that a sign of a term in $\det A$ is related to that of a factor in $D(A)$ by the following.

$$\det A = (-1)^n \sum_{k=1}^{q} (-1)^{m_k} A[f_k], \tag{2}$$

where k, q, and m_k have the same meanings as used in the description for (1) in section 2, $A[f_k]$ is a product of all elements contained in the factor f_k.

[Proof] Sufficiency: First, suppose that the arc (ij) satisfies the condition i). Let c_n and c_p denote a negative and positive cycles that include the arc (ij), and l_n, l_p be the length of c_n and c_p respectively. Suppose a factor of $D(A)$ which consists only of loops. From the factor, we can construct another factor by substituting l_n loops with c_n. The sign of factors does not change because, in view of (2), the sign of factors are $(-1)^{2n}$ and $(-1)^{2(n-l_n+1)}$ respectively, both of which are positive. Similarly, another factor can be constructed by substituting l_p loops with the c_p. In this case, however, the sign changes to $(-1)^{n-l_p+1} \times (-1)^{n-l_p}$, which is negative. Hence, we conclude that there exist two terms whose signs are opposite each other and both contain a_{ij}.

Next, suppose that the arc satisfies the condition ii). By substituting the positive cycle with loops, we can change the sign of a factor. Hence, the sufficiency is proved.

Necessity: Assume that neither of the two conditions holds. Then all cycles to which the arc belongs are of the same sign, and cycles disjoint to them are all negative. If the arc belongs to negative cycles, all cycles in $D(A^I)$ are negative, and there is no sign-conflict arc. If cycles to which the arc belongs are all positive, any factor which contains the arc must consists of one positive cycle and other negative cycles. Let l_p be the length of the positive cycle. Then the sign of the factor is $(-1)^{n-l_n+1} \times (-1)^{n-l_n}$, which is always negative. Hence, the necessity of the theorem holds. □

Obviously, from Lemma 3.1, the sign regular matrix contain no sign-conflict elements. An element which is not a sign-conflict type can be fixed at either of the terminals of its interval. Which terminal to choose depends on the sign of the cycle, the size of a matrix and the sign of the element itself. The next theorem shows their contribution to the determinant.

Theorem 3.2 *Assume that an interval* $a^I = [\underline{a}, \overline{a}]$ *of* A^I *is not a sign-conflict type. Let* $a_c = \frac{1}{2}(\underline{a} + \overline{a})$, $\delta = \frac{1}{2}(\overline{a} - \underline{a})$. *Further, let* c_a *denotes one of cycles in which* a^I *is contained and* $sgn\ c_a$ *be the sign of the cycle* c_a.

The determinant of A^I *is maximized with respect to* a^I *when the interval* a^I *is fixed at* $a_c + (-1)^{n+1} \times sgn\ c_a \times sgn\ a^I \times \delta$ *and minimized when it is fixed at the other terminal.*

[Proof] The sign of the factors in which a^I is contained can be uniquely determined: i.e. $-sgn\ c_a$. Hence, the sign of the corresponding term is $(-1)^n \times (-sgn\ c)$ in view of (2). Further, it is obvious that change in a^I linearly contributes to the determinant value. Hence, taking into account the contribution of $sgn\ a^I$, the result follows. □

If the interval matrix is sign regular, all the elements can be fixed by the rule in the above theorem, and the minimum and the maximum values of the determinant can be calculated. Even if some elements remains unfixed, the sufficient condition might apply because the radius decreases in some of its elements after fixing.

Based on the above results, we next present our algorithm to check regularity of interval matrices.

Algorithm 3.1

Step 1. *Transform a given interval matrix* A^I *to its standard form.*
Step 2. *Find sign-conflict arcs by checking signs of cycles in* $D(A)$.
Step 3. *Fix non sign-conflict elements at either of its terminals according to the sign of* $\det A_c$.
Step 4. *If* $\rho(|A_c^{-1}|\Delta) < 1$, *terminate.* A^I *is regular.*
Step 5. *Perform combinatorial computation. If determinants of all the vertex matrices are of the same sign,* A^I *is regular.* □

If the matrix can not be transformed to the standard form in Step 1, the matrix is singular. Note that the more elements are fixed in Step 3, the more combinatorial computation in Step 5 can be reduced.

Example 2:

Consider 3 × 3 interval matrix A^I shown below which has already been transformed to the standard form. Lemma 2.1 can not apply to A^I because $\rho(|A_c^{-1}|\Delta) = 1.4$. First, by examining the corresponding signed digraph (Figure 3), we can find two sign-conflict elements indicated by $*$. a_{13} is a sign-conflict type because the arc (13) in the digraph is contained in both the negative cycle (1321) and the positive cycle (131), where the condition i) of the Theorem 3.1 is met. a_{22} is also a sign-conflict one because the loop (22) is a cycle disjoint to the positive cycle (131), where the condition ii) is met. Other elements can be fixed at the underlined terminals according to the rule in Theorem 3.2. Note that because $\det A_c < 0$, we want to maximize the determinat to see if it crosses zero. As a result, we get a new interval matrix B^I which consists of both values and intervals. Observing that *det* B_c is still negative, the regularity of A^I is reduced to that of more restrictive interval matrices B^I.

$$A^I = \begin{pmatrix} [-6, \underline{-1}\,] & [-7, \underline{-1}\,] & *[\ 2,\ \ 3\,] \\ [\ \underline{8},\ 10\,] & *[-3, -2\,] & [\ 1,\ \ 6\,] \\ [\ 1,\ \underline{3}\,] & [-3, \underline{-1}] & [-4, \underline{-1}\,] \end{pmatrix}$$

$$B^I = \begin{pmatrix} -1 & -1 & *[\,2,\ 3\,] \\ 8 & *[-3, -2] & 1 \\ 3 & -1 & -1 \end{pmatrix}$$

Table 1: Sample matrices

No.	Size	Nonzero elements	Cycles	Conflicts	Regularity
1	5	25	89	25	regular
2	5	15	10	5	singular
3	8	22	15	0	regular
4	8	39	104	9	regular
5	14	42	30	0	regular
6	14	42	30	37	regular
7	14	82	69	0	regular
8	14	118	105	0	regular
9	14	109	105	0	regular
10	14	44	35	18	regular
11	15	54	486	53	singular
12	20	60	42	40	regular
13	20	62	47	59	regular
14	20	94	47	94	regular
15	20	72	1178	71	singular
16	20	82	4802	81	regular

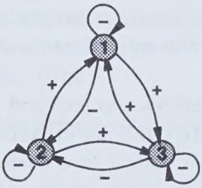


Figure 2: The digraph of Example 2

By fixing some intervals at their terminals, the radius Δ is expected to be smaller, which leads to an increase of applicability of the sufficient condition mentioned before, i.e., $\rho(D) < 1$. In the example shown above, $\rho(D) = 1.4$ for A^I, whereas $\rho(D) = 0.6$ for B^I. In this case, just fixing the non-sign-conflict elements allows us to judge the regularity of A^I. □

4 Experiments

We apply our method to Some sample matrices. Table 1 shows the number of nonzero elements, cycles in digraphs, sign-conflict elements and the size of sample matrices are shown with their regularities.

First we compare our method with Rohn's method [5], which is based on the combinatorial computation. Table 2 shows the processing time to judge the regularity. The result for our method is shown in column A, and Rohn's method in column B. Our method

Table 2: Processing time

		(sec)	
No.		A	B
1		7,360	0
2	F	0	0
3		0.1	0.1
4		0.1	0.1
5		0.2	5.5
6		0.2	6.8
7		8.9	9.0
8		11.9	5.7
9		7.3	5.1
10		0.3	6.4
11		1.3	17.4
12		0.4	904
13		0.4	1,051
14	F	60.2	1,174
15		7.4	1,383
16		81.5	930

Table 3: Change in $\rho(D)$

	$\rho(D)$	
No.		After fixing
1	1.412	1.412
2	1.224	1.116
3	1.218	0
4	1.203	0.532
5	1.831	0
6	1.245	0.986
7	1.489	0
8	1.457	0
9	1.618	0
10	1.962	0.825
11	18.31	7.292
12	0.986	0.533
13	1.027	0.703
14	1.875	1.875
15	20.75	35.11
16	1.045	0.995

may fail if the sufficient condition is not satisfied even after fixing non-conflict intervals. Such cases are denoted by F in Table 2.

As expected in Rohn's method, the processing time increases exponentially as the matrix size increases. On the other hand, in our method, the time seems to depends on the size of matrices, the number of non-zero elements and cycles. However the dependence is not exponential.

Compared with the Rohn's method, our method considers a interval matrix as the combination of the sign regular matrix and disturbances, i.e., sign-conflict elements, while Rohn's method considers the center matrix as a basis and radius as the disturbance.

Table 3 shows the change in applicability of the sufficient condition. The $\rho(D)$ values before and after fixing elements are shown. In most cases, the value after fixing is smaller than that before fixing. Especially, when there is no sign-conflict elements as in $3, 5, 7, 8, 9$, the value is equal to 0 after fixing. On the other hand, if all the elements are sign-conflict type as in $1, 14$, the value does not change. $\rho(D)$ is possible to increase as in 15.

From these observations, it is presented that our method would be effectively applicable for the cases such that the matrix is sparse and the number of conflicts is small.

5 Concluding remarks

Computations about interval matrices often require a computationally expensive combinatorial search. We have presented a new method to check the regularity of interval matrices. Utilizing the results so far derived for the sign matrices, we have obtained graph theoretical conditions to test whether a certain element can be fixed at one of its terminal values for minimizing/maximizing the determinant. We have shown examples that demonstrate the computation can be reduced. In most cases, the applicability of the sufficient condition is increased after fixing elements. Our algorithm can be applied before carrying out any other algorithms for checking regularity of interval matrices.

It is expected that our approach of using qualitative and structural information to reduce computations of interval matrices may be applied to many other similar problems: solvability of static interval systems, stability, initial value problem of dynamic interval systems, and so on. These problems are currently under investigation.

References

[1] J. Quirk and R. Ruppert. "Qualitative economics and stability of equilibrium". *Rev. Economic Studies*, 32:311–326, 1965.

[2] C. Jefferies. "Qualitative stability and digraph in model ecosystem". *Ecology*, 55:1415–1419, 1974.

[3] J. S. Maybee. "Sign Solvability". In H. Greenberg and J.S. Maybee, editors, *Proc. Symposium on Computer Assisted Analysis and Model Simplification.* Academic Press, New York, 1981.

[4] Y. Ishida, N. Adachi, and H. Tokumaru. "Some Results on the Qualitative Theory of Matrix". *Trans. of the Society of Instrument and Control Engineers*, 17:49–55, 1981.

[5] J. Rohn. "Systems of linear interval equations". *Linear Algebra and Its Applications*, 126:39–78, 1989.

[6] H. Beeck. "Zur Problematik der Hüllenbestimmung von Intervallgleichungssystemen". In K. Nickel, editor, *Interval Matrics*, Lecture Notes in Computer Science 29, pages 150–159. Springer-Verlag, Berlin, 1975.

[7] S. Poljak and J. Rohn. "Checking Robust Nonsingularity is NP-Hard". *Math. Control Signals Systems*, 6:1–9, 1993.

A Fixed-Slope Universal Sequential Algorithm for Lossy Source Coding Based on Gold-Washing Mechanism*

Qingchong Liu, Enhui Yang, and Zhen Zhang

Communication Sciences Institute
Department of Electrical Engineering-Systems
University of Southern California
Los Angeles, CA 90089-2565

Abstract

A novel fixed-slope universal sequential algorithm for lossy source coding based on Gold-Washing mechanism is suggested. Both the encoder and the decoder have two codebooks, a candidate codebook and a permanent codebook, which are updated each time according to the Gold-Washing mechanism. The candidate codebook contains the newly introduced codewords from the latest history of the output sequence. The permanent codebook holds the codewords which have shown good behavior in the past. The encoder always picks up the best codeword in the two codebooks to minimize a cost function, and outputs the Elias code of its index. This algorithm is applied to a binary independent, identically distributed source. The rate-distortion pairs returned are much closer to the fundamental rate-distortion curve than those obtained by applying the extended Lempel-Ziv-Welch (ELZW) algorithm suggested by Morita and Kobayashi. Our algorithm also converges faster than ELZW, while our codebook size is only 5% of that used by Morita and Kobayashi.

1 Introduction

The Gold-Washing(GW) mechanism of codebook refinement has been proven to be a powerful tool in on-line adaptive source coding both theoretically [1-2] and empirically [3]. In [1], Zhang and Wei proved that the performance of an on-line universal data compression algorithm based on GW can asymptotically approach the fundamental rate distortion limit of a finite discrete memoryless source with unknown statistics. In [2], Zhang and Yang proposed two new versions of the lossy data compression algorithm based on GW. They also proved that the performance of these two new versions of their algorithm, converges to the rate-distortion function when applied to a stationary, ergodic source satisfying uniformly strong mixing condition, or almost surely to the rate-distortion function when applied to a Markov source or a finite-state source. Chen [3]

---

*Supported by NSF under Grants NCR-9205265 and NCR-9508282.

showed that algorithms in image coding based on GW are superior to both VQ and JPEG. From the viewpoint of rate distortion theory, unfortunately, all of these algorithms are block code-based algorithms and, therefore, can not achieve the rate distortion limit in one pass, that is, without letting the block length of the block code go to infinity.

In this paper, we propose a fixed-slope universal sequential algorithm for lossy source coding based on the Gold-Washing mechanism (USGW). In contrast to the algorithms mentioned in the preceding paragraph, this algorithm is conjectured to achieve the rate distortion limit in one pass. Instead of pursuing the proof of the conjecture, in the following, we are mainly concerned with the application of this algorithm to a binary independent, identically distributed (i.i.d.) source and the corresponding comparison with the extended Lempel-Ziv-Welch (ELZW) algorithm suggested by Morita and Kobayashi [4]. We will leave the theoretic version of this algorithm and its corresponding analysis elsewhere.

Consider a source sequence $x = \{x_k\}_{k=0}^{K}$ drawn from an alphabet $\mathcal{A}$ with size N, where $\mathcal{A} = \{a_0, a_1, ..., a_{N-1}\}$. Assume that $\underline{x}$ is a subsequence of x, and $\underline{c}$ a codeword of the same length as that of $\underline{x}$, i.e., $|\underline{c}| = |\underline{x}|$. Define the cost of encoding $\underline{x}$ by $\underline{c}$ as

$$cost(\underline{c}, \underline{x}) = \frac{elias(index(\underline{c}))}{|\underline{c}|} + \lambda \frac{d(\underline{c}, \underline{x})}{|\underline{c}|} \tag{1}$$

where $index(\underline{c})$ is the index of the codeword $\underline{c}$, $index(\underline{c}) \in Z^+$, $elias(index(\underline{c}))$ the length of the Elias code [5] of the integer $index(\underline{c})$, $d(\underline{c}, \underline{x})$ the distortion between $\underline{x}$ and $\underline{c}$, and $\lambda > 0$ a constant which is meaningful as discussed in section 3. The function $elias(n)$ is defined [5] as

$$elias(n) = 1 + \lfloor log_2(n) \rfloor + 2\lfloor log_2(1 + \lfloor log_2(n) \rfloor) \rfloor \tag{2}$$

where $n \in Z^+$, and $\lfloor log_2(n) \rfloor$ gives the maximal integer not greater than $log_2(n)$. The definition of the distortion function $d(\underline{c}, \underline{x})$ in (1) is application dependent, because different applications may require different distortion measures. Therefore, the cost function (1) can be regarded as a summation of the rate and a weighted average distortion to encode a subsequence $\underline{x}$ by a codeword $\underline{c}$.

In order to use the Gold-Washing mechanism for codebook refinement, a threshold function is required [1-2]. In [1], it was conjectured that a threshold function in the order of n^ϵ should be enough, where n is codeword length and $\epsilon > 0$ is arbitrary. Our simulations show that a looser condition often bring some gain for a binary i.i.d. source sequence of finite length as shown in section 3 of this paper. A practical threshold function $threshold(n)$ can be defined as

$$threshold(n) = w - \lfloor log_N n \rfloor \tag{3}$$

where n is a codeword length, and w a constant. The constant w can be chosen in such a way that

$$\lim_{n \to n_{max}} threshold(n) = b \tag{4}$$

where n_{max} is the maximal codeword length, and b is a constant positive integer.

2 The Algorithm

The structure of our sequential data compression system is illustrated in Fig. 1. Both the encoder and the decoder use two codebooks, denoted by $CB0$ and $CB1$. The first

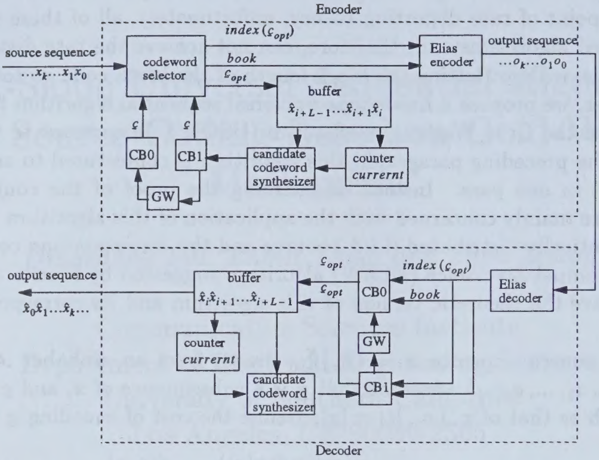

Encoder
index (c_{opt})
source sequence
…x_k…x_1x_0
codeword selector
book
c_{opt}
Elias encoder
output sequence
…o_k…o_1o_0
buffer
$\hat{x}_{i+L-1}…\hat{x}_{i+1}\hat{x}_i$
c
c
CB0
CB1
candidate codeword synthesizer
counter currernt
GW
c_{opt}
index (c_{opt})
output sequence
buffer
$\hat{x}_i\hat{x}_{i+1}…\hat{x}_{i+L-1}$
$\hat{x}_0\hat{x}_1…\hat{x}_k…$
c_{opt}
CB0
book
Elias decoder
GW
counter currernt
candidate codeword synthesizer
CB1
Decoder

Figure 1: Structure of the USGW encoder and decoder.

codebook $CB0$ aims at gathering codewords which have good performances in the past and the second codebook provides candidate codewords [2]. Both $CB0$ and $CB1$ are treated as lists, i.e., any codeword $\underline{c}$ in $CB0$ or $CB1$ can be reached through its index denoted by $index(\underline{c})$, which takes an integer value. In $CB1$, each codeword has a counter $q(\underline{c})$ to remember the times it has been used in the past. The gold-washer promotes good codewords from $CB1$ into $CB0$. In order to introduce candidate codewords, we need a buffer of length $L > 0$ to keep a recent history of the output sequence, a counter *current* to remember the number of symbols encoded or decoded to the present time, and a synthesizer to generate candidate codewords from the recent history. Hereafter, the content of the buffer is denoted by a sequence $\underline{h}$. The Elias encoder and decoder are used to encode and decode integers [5]. The codeword selector in the encoder is to pick up the best codeword in $\mathcal{C} = \{CB0\} \cup \{CB1\}$ which minimizes the cost function defined in (1).

Initially, let the codebook $CB0$ be empty, $CB1 = \mathcal{A}$, $current = 0$, and reset the buffers. This initialization is common to both the encoder and the decoder. To encode, the codeword selector of the encoder finds the best codeword $\underline{c}_{opt}$ in $\mathcal{C} = \{CB0\} \cup \{CB1\}$ and the best subsequence $\underline{x}_{opt}$ of the input such that

$$cost(\underline{c}_{opt}, \underline{x}_{opt}) = \min_{\underline{c} \in \mathcal{C}} cost(\underline{c}, \underline{x}) \tag{5}$$

where $\underline{x} = x_i x_{i+1} ... x_{i+l-1}$ is a subsequence of the input, $i = current$, and $l = |\underline{c}|$. If more than one codewords satisfy the condition (5), choose the longest one. Then the codeword selector outputs one bit *book* and the index of $\underline{c}_{opt}$ to the Elias encoder, and $\underline{c}_{opt}$ to the buffer. The variable *book* is defined as

$$book = \begin{cases} 0 & \text{if } \underline{c}_{opt} \in CB0 \\ 1 & \text{otherwise} \end{cases}$$

The Elias encoder outputs *book* followed by the Elias code of $index(\underline{c}_{opt})$. Then, the number $|\underline{c}|$ is added to *current*. Hence, the input subsequence $\underline{x}_{opt}$ has been encoded by the codeword $\underline{c}_{opt}$. Each time the buffer receives a codeword $\underline{c}_{opt}$, it appends $\underline{c}_{opt}$ to the end of the buffer sequence $\underline{h}$. If this appending will lead to an overflow of the buffer, a prefix of $\underline{h}$ should be deleted before doing the appending, so that the buffer would have enough space to hold $\underline{c}_{opt}$. The candidate codeword synthesizer parses the buffer sequence $\underline{h}$ from the $(|\underline{h}| - |\underline{c}_{opt}| - k)$-th symbol on into phrases of length k, $1 \leq k \leq \lfloor log_2(current - |\underline{c}_{opt}| + 3\rfloor$. If any of these phrases is not in $\mathcal{C}$, add it into $CB1$ and set its counter zero. Furthermore, if $\lfloor log_2(current - |\underline{c}_{opt}| + 3\rfloor < \lfloor log_2(current + 3)\rfloor$, then parse $\underline{h}$ into phrases of length $\lfloor log_2(current + 3)\rfloor$, add all of them into $CB1$, and set their counters zeroes. The gold-washing mechanism works in the following way: If $book = 0$, just move $\underline{c}_{opt}$ to the front of $CB0$. Otherwise, add 1 to the counter of $\underline{c}_{opt}$. If the counter of $\underline{c}_{opt}$ is greater than $threshold(|\underline{c}_{opt}|)$, promote $\underline{c}_{opt}$ to $CB0$ and delete it from $CB1$; Otherwise, just move $\underline{c}_{opt}$ to the front of $CB1$. This process is repeated in the encoder until the end of the input sequence is reached.

At the decoder, the Elias decoder recovers the variable *book* and the index $index(\underline{c}_{opt})$ first. If $book = 0$, the $index(\underline{c}_{opt})$-th codeword of $CB0$ is sent to the output buffer; otherwise, the $index(\underline{c}_{opt})$-th codeword of $CB1$ is sent. The counter, the candidate codeword synthesizer, and the gold-washing mechanism work in the same way as in the encoder. The decoder repeats the same process until the end of the input is reached.

Now our algorithm can be described as the following pseudocode:

Encoding

```
/* Initialization: */
  Let $CB0$ be empty, $CB1 = \mathcal{A}$;
  Let the buffer be empty;
  $current = 0$;
do {
   find $\underline{c}_{opt}$ and $\underline{x}_{opt}$ satisfying (5);
   /* Encode $\underline{x}_{opt}$ by $\underline{c}_{opt}$: */
     if ($\underline{c}_{opt} \in CB0$)  output $book = 0$;
     else  output $book = 1$;
     output the Elias code of $index(\underline{c}_{opt})$;
     $current = current + |\underline{c}_{opt}|$;
   /* Adjust the buffer sequence $\underline{h}$: */
     if ($|\underline{h}| + |\underline{c}_{opt}| > L$)  delete the first $(|\underline{h}| + |\underline{c}_{opt}| - L)$ symbols of $\underline{h}$;
     append $\underline{c}_{opt}$ to the end of $\underline{h}$;
   /* Adjust the codebooks: */
     if ($\underline{c}_{opt} \in CB0$)  move $\underline{c}_{opt}$ to the front of $CB0$;
     else {
       $q(\underline{c}_{opt}) = q(\underline{c}_{opt}) + 1$;
       if ($q(\underline{c}_{opt}) > threshold(|\underline{c}_{opt}|)$)  {
         add $\underline{c}_{opt}$ into $CB0$;
         delete $\underline{c}_{opt}$ from $CB1$;
         }
       else  move $\underline{c}_{opt}$ to the front of $CB1$;
     }
   /* Introduce new codewords: */
     for ($k = 1; k \leq \lfloor log_2(current - |\underline{c}_{opt}| + 3)\rfloor; k++$)  {
            for   $(j = |\underline{h}| - |\underline{c}_{opt}| - k; j \leq |\underline{h}| - k; j++)$   {
              if   $(\underline{d} = h_j h_{j+1} ... h_{j+k-1} \notin \mathcal{C})$   {
                add $\underline{d}$ into $CB1$;
                $q(\underline{d}) = 0$;
              }
            }
          }
        if   $(\lfloor log_2(current - |\underline{c}_{opt}| + 3)\rfloor < k = \lfloor log_2(current + 3)\rfloor)$   {
            for   $(j = 0; j \leq |\underline{h}| - k; j++)$   {
              add $\underline{d} = h_j h_{j+1} ... h_{j+k-1}$ into $CB1$;
              $q(\underline{d}) = 0$;
            }
          }
    } while (not reached the end of input sequence);
```

Decoding

```
    /* Initialization: */
      Let $CB0$ be empty, $CB1 = \mathcal{A}$;
      Let the buffer be empty;
      $current = 0$;
    do   {
      /* Recover book and $index(\underline{c}_{opt})$: */
        read one bit book from the input;
        recover an integer $index(\underline{c}_{opt})$ from the input[1];
      /* Output a codeword: */
        if   $(book == 0)$   let $\underline{c}_{opt}$ be the $index(\underline{c}_{opt})$-th codeword of $CB0$;
        else   let $\underline{c}_{opt}$ be the $index(\underline{c}_{opt})$-th codeword of $CB1$;
        output $\underline{c}_{opt}$;
        $current = current + |\underline{c}_{opt}|$;
      /* Adjust the buffer sequence $\underline{h}$: */
        if   $(|\underline{h}| + |\underline{c}_{opt}| > L)$   delete the first $(|\underline{h}| + |\underline{c}_{opt}| - L)$ symbols of $\underline{h}$;
        append $\underline{c}_{opt}$ to the end of $\underline{h}$;
      /* Adjust the codebooks: */
        if   $(\underline{c}_{opt} \in CB0)$   move $\underline{c}_{opt}$ to the front of $CB0$;
        else   {
          $q(\underline{c}_{opt}) = q(\underline{c}_{opt}) + 1$;
          if   $(q(\underline{c}_{opt}) > threshold(|\underline{c}_{opt}|))$   {
            add $\underline{c}_{opt}$ into $CB0$;
            delete $\underline{c}_{opt}$ from $CB1$;
            }
          else   move $\underline{c}_{opt}$ to the front of $CB1$;
        }
      /* Introduce new codewords: */
        for   $(k = 1; k \leq \lfloor log_2(current - |\underline{c}_{opt}| + 3)\rfloor; k++)$   {
          for   $(j = |\underline{h}| - |\underline{c}_{opt}| - k; j \leq |\underline{h}| - k; j++)$   {
            if   $(\underline{d} = h_j h_{j+1} ... h_{j+k-1} \notin \mathcal{C})$   {
              add $\underline{d}$ into $CB1$;
```

---

[1] Use Elias decoding method[5].

```
          $q(\underline{d}) = 0;$
          }
      }
    }
    if  $(\lfloor log_2(current - |\underline{c}_{opt}| + 3)\rfloor < k = \lfloor log_2(current + 3)\rfloor)$  {
      for  $(j = 0; j \leq |\underline{h}| - k; j++)$  {
        add $\underline{d} = h_j h_{j+1} ... h_{j+k-1}$ into $CB1$;
        $q(\underline{d}) = 0;$
      }
    }
}  while (not reached the end of input sequence);
```

3 Application to a Binary Source

For a binary source, the distortion function $d(\underline{c}, \underline{x})$ in (1) can be chosen as the Hamming distance. Then the absolute distortion is the Hamming distance between the source sequence and the decoded sequence. The normalized distortion d, usually called the distortion, is the absolute distortion divided by the length of the source sequence in bits. The rate r is the length in bits of the output sequence of the encoder divided by the length of the source sequence. If the source sequence is binary i.i.d., a meaningful choice of the parameter λ in (1) is the negative of the slope of the rate-distortion curve of the source. This choice makes the algorithm to return a pair (d, r) of distortion and rate located in a near neighborhood of the point on the rate-distortion curve with a slope of negative λ.

This algorithm is applied to a binary i.i.d. source which takes 0's with a probability $p = .2$, and 1's with probability $(1 - p) = .8$. This distribution is chosen purely to be consistent with that used in [4] for a comparison. The length $|x|$ of the source file is 16384 bits. The two codebook sizes are $|CB0| = 64$ and $|CB1| = 128$, respectively. A buffer of size $L = 256$ bits is used. The threshold function used is defined in (3) and (4) with $(N, b) = (2, 2)$. The returned distortions and rates for arbitrarily chosen λ's are listed in Table 1. The simulation result of applying ELZW [4] to a binary i.i.d. source of length $|x| = 32768$ bits with the same probability distribution is listed in Table 2. Three crucial parameters of ELZW are the codebook size $|CB|$, the number M of intermediate strings allowed, and the size P of a candidate table [4]. In our simulation, $(|CB|, M, P) = (4096, 16, 32)$ is used, which satisfies the best condition to get the results in [4]. Figure 2 is the plots for the data in table 1 and table 2 and the fundamental rate-distortion curve for the specified binary i.i.d. source. Figure 3 is the plots of USGW algorithm for different codebook sizes to the same source. Figure 4 is USGW with two different threshold functions, (3) and

$$threshold(n) = \lfloor k * log_N\, n \rfloor + w \tag{6}$$

where (k, N, w) is chosen as $(.3, 2, 2)$.

From our simulation results for both of our USGW algorithm and the ELZW algorithm, we can draw at least the following conclusions:

1. USGW algorithm converges faster than the ELZW. This can be seen from Fig. 2, in which USGW brought lower rates even for a shorter source sequence.

Table 1: Rates and distortions of applying USGW algorithm to a binary i.i.d. source with $(|x|, |CB0|,, |CB1|, L) = (16384, 64, 128, 256)$.

distortion(D)	.025672	.059353	.079002	.094739
rate(R)	.862132	.689655	.582614	.509410
distortion(D)	.110565	.136745	.156978	.189423
rate(R)	.443520	.335378	.272849	.200043

Table 2: Rates and distortions of applying ELZW algorithm to a binary i.i.d. source with $(|x|, |CB|, M, P) = (32768, 4096, 16, 32)$.

distortion(D)	.006195	.0367130	.049347	.105774	.115021
rate(R)	1.024292	.856567	.795776	.572754	.539795
distortion(D)	.133789	.157257	.174469	.190277	
rate(R)	.486328	.422241	.375366	.334351	

2. USGW algorithm does not need a big codebook (Fig. 3), while a big codebook is essential in ELZW algorithm. A big codebook size will essentially increase the complexity of these two algorithms, and also demand more memory in practice. Since in ELZW, a binary source was grouped into bytes [4], i.e., $|\mathcal{A}| = 2^8$, a codebook size less than several thousands was meaningless, as shown by our simulation. This problem can be solved by adding the gold-washing mechanism into ELZW.

3. USGW has a better control of the distortion than the ELZW.

4 Conclusion

A Fixed-slope universal sequential algorithm based on the gold-washing mechanism for lossy data compression has been suggested. Simulations have shown that it is better than the extended Lempel-Ziv-Welch algorithm [4] when applied to a binary i.i.d. source in the convergent speed, the demand of codebook size, and the distortion controlibility. Further work needs to be done to analyze these two algorithms.

References

[1] Z. Zhang and V. Wei, "An On-line Universal Lossy Data Compression Algorithm via Continuous Codebook Refinement", to appear in *IEEE Trans. Information Theory*

[2] Z. Zhang and E. Yang, "An On-line Universal Lossy Data Compression Algorithm via Continuous Codebook Refinement-Part II: Optimality for Stationary Sources

Satisfying Uniformly Strong Mixing Condition", to appear in *IEEE Trans. Information Theory*

[3] W. S. Chen, "Adaptive Vector Quantization for Image Compression: Algorithms, Performances, and VLSI Architecture", PhD thesis, Unvi. of Sourthen California, Sept. 1994

[4] H. Morita and K. Kobayashi, "An Extension of LZW Algorithm to Source Coding Subject to a Fidelity Criterion", Fourth Joint Swedish-Soviet International Workshop on Information Theory, Gotland, Sweden, August 1989

[5] P. Elias, "Universal Codeword Sets and Representations of Integers," *IEEE Trans. Inform. Theory*, vol. IT-21, no. 2, pp. 194-203, March 1975.

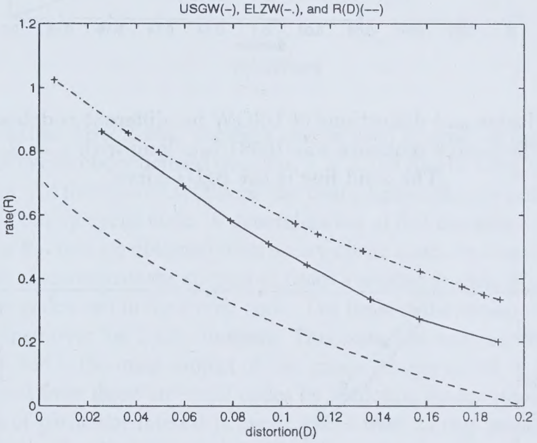


Figure 2: Rates and distortions of applying USGW algorithm with $(|x|, |CB0|, |CB1|, L) = (16384, 64, 128, 256)$ and the ELZW algorithm with $(|x|, |CB|, M, P) = (32768, 4096, 16, 32)$ to a binary i.i.d. source of $p = .2$

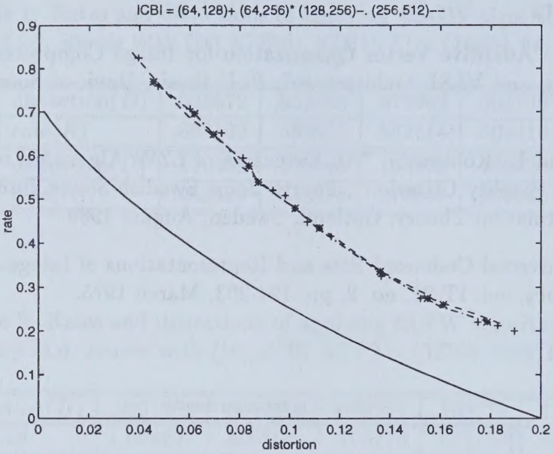


Figure 3: Rates and distortions of USGW for different codebook sizes.
The source sequence was 16384 bits long with $p = .2$.
The solid line is the R(D) curve.

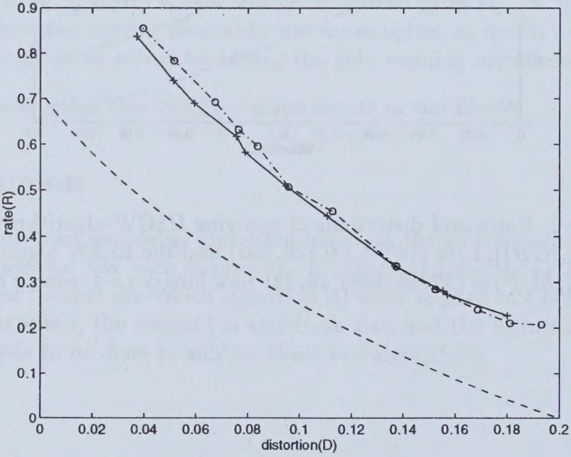


Figure 4: Rates and distortions of USGW with different threshold functions.
Solid line: threshold function (3) with $(N, b) = (2, 2)$;
Dash-dot line: threshold function (6) with $(k, N, w) = (.3, 2, 2)$;
Dash line: $R(D)$ curve with $p = .2$.

A 2-ADIC APPROACH TO THE ANALYSIS OF CYCLIC CODES

A. R. Calderbank
Mathematical Sciences Research Center
AT&T Bell Laboratories
Murray Hill, New Jersey 07974

Wen-Ching Winnie Li*,†
Department of Mathematics
Penn. State University
University Park, Pennsylvania 16802

Bjorn Poonen*,‡
Mathematical Sciences Research Institute
Berkeley, California 94720

Abstract

This paper describes how 2-adic numbers can be used to analyze the structure of binary cyclic codes and of cyclic codes defined over $\mathbb{Z}_{2^a}$, $a \geq 2$, the ring of integers modulo 2^a. It provides a 2-adic proof of a theorem of McEliece that characterizes the possible Hamming weights that can appear in a binary cyclic code. A generalization of this theorem is derived that applies to cyclic codes over $\mathbb{Z}_{2^a}$ that are obtained from binary cyclic codes by a sequence of Hensel lifts. This generalization characterizes the number of times a residue modulo 2^a appears as a component of an arbitrary codeword in the cyclic code. The limit of the sequence of Hensel lifts is a universal code defined over the 2-adic integers. This code was first introduced by Calderbank and Sloane (1995), and is the main subject of this paper. Binary cyclic codes and cyclic codes over $\mathbb{Z}_{2^a}$ are obtained from these universal codes by reduction modulo some power of 2.

A special case of particular interest is cyclic codes over $\mathbb{Z}_4$ that are obtained from binary cyclic codes by means of a single Hensel lift. The binary images of such codes under the Gray isometry include the Kerdock, Preparata and Delsarte-Goethals codes. These are nonlinear binary codes that contain more codewords than any linear code presently known. Fundamental understanding of the composition of codewords in cyclic codes over $\mathbb{Z}_4$ is central to the search for more families of optimal codes. This paper also constructs even unimodular lattices from the Hensel lift of extended binary cyclic codes that are self-dual with all Hamming weights divisible by 4. The Leech lattice arises in this way as do extremal lattices in dimensions 32 through 48.

1. Introduction

The purpose of introducing 2-adic methods to algebraic coding theory is to shed new light on cyclic codes defined over the binary field and over $\mathbb{Z}_{2^a}$, the ring of integers modulo 2^a.

---

[*] Research supported in part by AT&T Bell Laboratories.

[†] Research supported in part by NSA Grant No. MDA904-95-H-1006.

[‡] Research supported in part by an NSF Mathematical Sciences Postdoctoral Research Fellowship. Research at MSRI is supported in part by NSF grant DMS-9022140. Current address: Department of Mathematics, Princeton University, Princeton, NJ 08544.

Binary cyclic codes correspond to ideals in the ring $\mathbb{F}_2[X]/(X^N - 1)$, where $\mathbb{F}_2$ denotes the binary field. A binary cyclic code C is generated by a divisor $g(X)$ of $X^N - 1$ in $\mathbb{F}_2[X]$. The nonzeros of C are the roots of the the polynomial $h(X) = (X^N - 1)/g(X)$ in some extension of the binary field $\mathbb{F}_2$. McEliece's Theorem (given here as Theorem 2.1) characterizes the possible Hamming weights that can appear in a binary cyclic code C in terms of ℓ, the smallest number such that ℓ nonzeros of C have product 1. This characterization is that all Hamming weights are divisible by $2^{\ell-1}$, and that some Hamming weight is not divisible by 2^ℓ. This paper provides a 2-adic proof of McEliece's Theorem and its generalization. The proof that all Hamming weights are divisible by $2^{\ell-1}$ is very short, and the key idea is to evaluate the limit of a Cauchy sequence of 2-adic integers in the 2-adic metric.

Hensel's Lemma allows us to refine a factorization $X^N - 1 = g_2(X)h_2(X)$ modulo 2 to a factorization $X^N - 1 = g_{2^a}(X)h_{2^a}(X)$ modulo 2^a and to a factorization $X^N - 1 = g(X)h(X)$ over the 2-adic integers. The polynomial $g_{2^a}(X)$ generates an ideal in $\mathbb{Z}_{2^a}[X]/(X^N - 1)$, corresponding to a cyclic code C_{2^a}, called the Hensel lift of the initial binary cyclic code generated by $g_2(X)$. Cyclic codes obtained by Hensel lifting can also be described in terms of parity checks involving Galois rings. This is completely analogous to the construction of binary cyclic codes by means of parity checks involving finite fields. The Galois ring is obtained from $\mathbb{Z}_{2^a}$ by adjoining the $(2^m - 1)$th roots of unity for suitable m. These roots of unity will include all the roots of $g_{2^a}(X)$.

The limit of the sequence of Hensel lifts is a universal code C_∞ defined over the 2-adic integers. At this point we recall that arbitrary cyclic codes over the 2-adic integers were first introduced by Calderbank and Sloane [4]. The particular universal code can be described in terms of parity checks involving a field K obtained from the 2-adic field $\mathbb{Q}_2$ as above. By reducing the description of the universal code modulo different powers of 2, we obtain the Galois ring description of cyclic codes defined over $\mathbb{Z}_{2^a}$.

Much of the interest in cyclic codes over the rings $\mathbb{Z}_{2^a}$ stems from the recent work by Hammons et al. [9] on the $\mathbb{Z}_4$-linearity of the Kerdock, Delsarte-Goethals and Preparata codes. These are families of nonlinear binary codes that contain more codewords than any comparable linear code presently known. The Kerdock and Preparata codes can be very simply constructed as binary images under the Gray isometry of cyclic codes over $\mathbb{Z}_4$ obtained by Hensel lifting (see [9]). This paper also provides an application of 2-adic methods to the construction of even unimodular lattices. These lattices are obtained via Construction A (see [6, Chapter 5]) applied to extended cyclic codes over $\mathbb{Z}_4$ obtained by Hensel lifting.

2. Cyclic Codes

We shall consider a number of different alphabets; $\mathbb{F}_2$ denotes the binary field, $\mathbb{Z}_2$ denotes the ring of 2-adic integers, and for $a \geq 2$, $\mathbb{Z}_{2^a} = \mathbb{Z}_2/(2^a\mathbb{Z}_2)$ denotes the ring of integers modulo 2^a. A *cyclic code* of length N over one of these alphabets A is an ideal in the ring $A[X]/(X^N - 1)$. Throughout this paper we assume N is odd. Multiplication by X in this ring corresponds to the cyclic shift which fixes the set of codewords. Codewords can be represented by residues modulo $X^N - 1$, that is by polynomials of degree at most $N - 1$. The Hamming weight of a codeword is the number of nonzero coefficients of the associated polynomial.

We begin with the binary alphabet $\mathbb{F}_2$. Here the ring $\mathbb{F}_2[X]/(X^N - 1)$ is principal and a cyclic code C is generated by a divisor $g(X)$ of $X^N - 1$ in $\mathbb{F}_2[X]$. We write $X^N - 1 = g(X)h(X)$, and define the *nonzeros* of C to be the roots of $h(X)$ in some field extension of $\mathbb{F}_2$. The following theorem of McEliece [11] characterizes the possible Hamming weights that can appear in a binary cyclic code.

Theorem 2.1 (McEliece) *Let C be a binary cyclic code, and let ℓ be the smallest number such that ℓ nonzeros of C (with repetitions allowed) have product 1. Then the weight of every codeword in C is divisible by $2^{\ell-1}$ and there is at least one weight which is not divisible by 2^{ℓ}.*

Example. Here $N = 2^m - 1$ and the binary cyclic code C is the simplex code where the Hamming weight of every nonzero codeword is 2^{m-1}. This code is generated by the reciprocal of $(X^N - 1)/(g(X))$ where $g(X)$ is a primitive irreducible polynomial of degree m. Any root ω of $g(X)$ is a primitive element for the finite field $\mathbb{F}_{2^m} = \mathbb{F}_2[\omega]$. The nonzeros of C are the conjugates of ω under the Frobenius map, namely $\omega, \omega^2, \omega^4, \ldots, \omega^{2^{m-1}}$. We have

$$\prod_{i=0}^{m-1} \omega^{2^i} = \omega^{1+2+4+\cdots+2^{m-1}} = \omega^{2^m-1} = 1$$

and it is evident that $\ell = m$. ■

The next result is an analog of McEliece's Theorem for cyclic codes that are defined over $\mathbb{Z}_4$, or more generally over the ring $\mathbb{Z}_{2^a}$.

Theorem 2.2. *Let C be a binary cyclic code, and let ℓ be the smallest number such that ℓ nonzeros of C (with repetitions allowed) have product 1. Then for any codeword in the Hensel lift C_{2^t} over $\mathbb{Z}_{2^t}$, and for any $j \in \mathbb{Z}_{2^t}$, the number of components equal to j is divisible by $2^{\lceil \ell/2^{t-1} \rceil - 2}$.*

3. Even Unimodular Lattices

At this point it is useful to consider the Euclidean weight of a vector in $\mathbb{Z}_4^N$. The *Euclidean weights* of 0, 1, 2, 3 in $\mathbb{Z}_4$ are 0, 1, 4, 1 respectively, and the Euclidean weight of a vector is the rational sum of the Euclidean weights of the components. More generally we define the Euclidean weight of the element i in $\mathbb{Z}_{2^a}$ to be $\min\{i^2, (2^a - i)^2\}$. The analog of McEliece's Theorem for cyclic codes over $\mathbb{Z}_4$ guarantees that the number of 2s in a codeword is divisible by a certain power of 2. It leaves open the possibility that the number of occurrences may be zero. The next theorem can be used to show that the number of 2s appearing in certain codewords must be nonzero.

Theorem 3.1. *Let C be a binary cyclic code of odd length N that is self-orthogonal. Then all Hamming weights in C are divisible by 4, and all Euclidean weights in the Hensel lift C_{2^a} are divisible by 2^{a+1}.*

Remarks. If $4|N + 1$ then we may obtain an extended binary cyclic code by appending an overall parity check to the cyclic code generated by C and the all-one vector of length N. This extended code is self-orthogonal with all Hamming weights divisible by 4.

If $2^{a+1}|N+1$ then we may obtain an extended cyclic code over $\mathbb{Z}_{2^a}$ by appending an overall parity check to the cyclic code generated by the Hensel lift C_{2^a} and the all-one vector of length N. This extended code is self-orthogonal with all Euclidean weights divisible by 2^{a+1}.

Bonnecaze, Calderbank and Solé [1] showed that certain self-dual codes over $\mathbb{Z}_4$ determine even unimodular lattices. The lattice $\Lambda(C_4)$ corresponding to a self-dual code C_4 in $\mathbb{Z}_4^N$ is given by

$$\Lambda(C_4) = \{x \in \mathbb{Z}^N \mid x \equiv c \pmod 4 \text{ for some } c \in C_4\}\ .$$

The fundamental volume of $\Lambda(C_4)$ is 2^N, since this is the index of C_4 in $\mathbb{Z}_4^N$. If $x_1, x_2 \in \Lambda(C_4)$ then $x_1 = c_1 + 4\omega_1$, $x_2 = c_2 + 4\omega_2$, where $\omega_1, \omega_2 \in \mathbb{Z}^N$ and $c_1, c_2 \in C_4$. Since C_4 is self-orthogonal,

$$(x_1, x_2) = (c_1 + 4\omega_1, c_2 + 4\omega_2) \equiv 0 \pmod 4$$

so that $\Lambda(C_4)/2$ is an integral lattice with determinant 1. It is easy to see that the condition that all norms in $\Lambda(C_4)/2$ are even is equivalent to the condition that all Euclidean weights in C_4 are divisible by 8.

More generally we may define a lattice $\Lambda(C_{2^a})$ corresponding to a self-dual code C_{2^a} in $\mathbb{Z}_{2^a}^N$ by

$$\Lambda(C_{2^a}) = \{x \in \mathbb{Z}^N | x \equiv c \pmod{2^a} \text{ for some } c \in C_{2^a}\}\ .$$

If all Euclidean weights in C_{2^a} are divisible by 2^{a+1} then $\Lambda(C_{2^a})/2^{a/2}$ is an even unimodular lattice.

Corollary 3.2. *Let C be an extended binary cyclic code of length $N+1 \equiv 0 \pmod{2^{a+1}}$ that is self-dual. If C_{2^a} is the Hensel lift of C, then $\Lambda(C_{2^a})/2^{a/2}$ is an even unimodular lattice.*

Remarks. Bonnecaze, Calderbank and Solé proved that the Hensel lifts of the extended binary quadratic residue codes of length $q+1$ ($q \equiv -1 (\text{mod } 8)$) determine even unimodular lattices in this way. The extended quadratic residue code of length 24 determines the Leech lattice, and this is perhaps the simplest construction for this remarkable lattice that is known. Instances of Corollary 3.2 lead to the construction of extremal type II lattices in dimensions 32, 40 and 48; for more details see Conway and Sloane [6, Chapter 7], Calderbank et al. [3], and Calderbank and Sloane [5].

All we are doing here is applying Construction A [6, Chapter 5] to linear codes defined over $\mathbb{Z}_{2^a}$. When $a = 2$, the minimum norm in $\Lambda(C_4)/2$ is at most 4 since $\Lambda(C_4)/2$ contains $2\mathbb{Z}^N$. A minor modification (Construction B) produces lattices containing $2D_N$ which has minimum norm 8. Hence this construction has the potential of constructing extremal lattices in dimensions up to 72. The same constructions applied to binary codes are not useful in dimensions beyond 40 since they produce even unimodular lattices where the minimum norm is at most 4. In general, the lattice $\Lambda(C_{2^a})/2^{a/2}$ contains $2^a\mathbb{Z}^N/2^{a/2}$, hence the minimum norm might be as large as 2^a.

Finally we remark that it can be shown that the even unimodular lattices produced by Construction A as above are good, in the sense that it is possible to achieve the Conway-Thompson lower bound.

Postscript

Stimulated by an earlier draft of this work, Wilson [15] has produced an alternative derivation of Theorem 2.2 that makes less essential use of the 2-adic integers. For example he uses Hensel's Lemma to produce 2-adic factorizations that are arbitrarily fine, without explicitly constructing the 2-adic limit. He also constructs mapping polynomials by making creative use of the calculus of finite differences.

Acknowledgements

A. R. Calderbank would like to thank Gary McGuire for helpful discussions regarding a non 2-adic proof of a weaker version of Theorem 3.1 which concerns Euclidean weights in codes over $\mathbb{Z}_4$ obtained by Hensel lifting. He would also like to thank the Institut des Hautes Etudes Scientifiques for support and hospitality during the preparation of this manuscript.

References

[1] A. Bonnecaze, A. R. Calderbank, and P. Solé, *Quaternary quadratic residue codes and unimodular lattices*, IEEE Trans. Inform. Theory, vol. 41, pp. 366–377, 1995.

[2] A. R. Calderbank and G. McGuire, *$\mathbb{Z}_4$-linear codes obtained as projections of Kerdock and Delsarte-Goethals codes*, to appear in Lin. Alg. and Appl.

[3] A. R. Calderbank, G. McGuire, P. V. Kumar and T. Helleseth, *Cyclic codes over $\mathbb{Z}_4$, locator polynomials and Newton's identities*, submitted to IEEE Trans. Inform. Theory.

[4] A. R. Calderbank and N. J. A. Sloane, *Modular and p-adic cyclic codes*, Designs, Codes and Cryptography, vol. 6, pp. 21–35, 1995.

[5] A. R. Calderbank and N. J. A. Sloane, *Double circulant codes over $\mathbb{Z}_4$ and even unimodular lattices*, submitted to J. Algebraic Combinatorics.

[6] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, 2nd ed. New York: Springer-Verlag, 1993.

[7] J. H. Conway and N. J. A. Sloane, *Quaternary constructions for the binary single error correcting codes of Julin, Best and others*, Designs, Codes and Cryptography, pp. 31–42, 1994.

[8] F. Q. Gouvêa, *p-adic Numbers*, Springer-Verlag, New York, 1993.

[9] A. R. Hammons, Jr., P. V. Kumar, A. R. Calderbank, N. J. A. Sloane and P. Solé, *The $\mathbb{Z}_4$-linearity of Kerdock, Preparata, Goethals and related codes*, IEEE Trans. Inform. Theory, vol. 40, pp. 301–319, 1994.

[10] A. M. Kerdock, *A class of low-rate nonlinear binary codes*, Inform. Contr., vol. 20, pp. 182–187, 1972.

[11] R. J. McEliece, *On periodic sequences from $GF(q)$*, J. Combinatorial Theory, vol. 10, pp. 80–91, 1971.

[12] V. Pless and Z. Qian, *Cyclic codes and quadratic residue codes over $\mathbb{Z}_4$*, preprint, 1995.

[13] F. P. Preparata, *A class of optimum nonlinear double error correcting codes*, Inform. and Control, vol. 13, pp. 378-400, 1968.

[14] T. Verhoeff, *An updated table of minimum distance bounds for binary linear codes*, IEEE Trans. Inform. Theory, vol. 39, pp. 662–667, 1993.

[15] R. M. Wilson, *A version for the Lee metric of a theorem of McEliece on weights of codewords in cyclic codes*, Lecture Notes, February 22, 1995.

On Galois Rings: A Hasse-Davenport-type Relation for Z_4 Kerdock Code and Applications

Tor Helleseth*, P. Vijay Kumar, Abhijit G. Shanbhag[†], and Kyeoncheol Yang[‡]

Abstract

In this talk, we obtain a Hasse-Davenport type of a relation for the *Kerdock exponential sum.* This relation can be used to obtain a closed-form expression for the Lee weight of a codeword in the Kerdock code in terms of its trace expansion. In addition, it is also shown that the generalized Hamming weights of longer Kerdock codes can sometimes be determined from those of shorter codes.

I. Introduction

The Kerdock code was shown by Hammons, Kumar, Calderbank, Sloane, Sole (IT., Mar '94) to be expressible as the image under the Gray map of a Z_4-linear trace code. The Lee distance distribution of the Kerdock code (and other Z_4-linear trace codes) is closely related to Weil-type exponential sums over Galois rings.

Let $GR(4,m)$ denote a Galois ring of size 4^m. Let $\beta \in GR(4,m)$ have order 2^m-1. Set

$$\mathcal{T}_m = \{0, 1, \beta, \beta^2, \cdots, \beta^{2^m-2}\}.$$

Every $z \in GR(4,m)$ has unique base-2 expansion

$$z = x + 2y, \ x, y \in \mathcal{T}_m.$$

The trace function $T_1^m(\cdot) : GR(4,m) \to Z_4$ is given by

$$T_1^m(z) = \sum_{i=0}^{m-1} (x^{2^i} + 2y^{2^i}).$$

II. Lee Weight Distribution of th Kerdock code via a Hasse-Davenport type of Relation

The Kerdock code is a $(2^{m+1}, 2^{2m+2}, 2^m - 2^{\frac{m-1}{2}})$, m odd, non-linear binary code. Every codeword in the Z_4 Kerdock code $\mathcal{K}$ is of the form

$$c + T_1^m([a+2b]x), \ a, b \in \mathcal{T}_m, \ c \in Z_4$$

and its Lee weight is

$$2^m - \Re\{i^c \sum_{x \in \mathcal{T}_m} i^{T_1^m([a+2b]x)}\}. \tag{1}$$

---

*Tor Helleseth is with the Department of Informatics, University of Bergen, Høyteknologisenteret, N-5020 Bergen, Norway.

[†] P. Vijay Kumar and Abhijit G. Shanbhag are with the Communication Sciences Institute, EE-Systems, EEB 534, University of Southern California, Los Angeles, CA 90089-2565, USA.

[‡] Kyeoncheol Yang is with Hanyang University, Seoul.

This work was supported in part by the Norwegian Research Council under Grant Numbers 107542/410 and 107623/420, the National Science Foundation under Grant Number NCR-93-05017 and in part by Korean Ministry of Science and Information.

Thus, for the Lee weight distribution, it suffices to compute the Kerdock exponential sum

$$\Gamma_m(a+2b) := \sum_{x\in\mathcal{T}_m} i^{T_1^m([a+2b]x)}.$$

We assume $a \neq 0$ since if $a = 0$,

$$\Gamma_m(0+2b) = 0.$$

When $a \neq 0$, we can assume $a = 1$ since by cyclically shifting

$$\Gamma_m(a+2b) = \Gamma_m(1+2\frac{b}{a}).$$

Further it can be shown that

$$\Gamma_m(1+2b) = i^{-T_1^m(b)}\Gamma_m(1)$$

and so we need to find only $\Gamma_m(1)$. For all $s, m \geq 1$, Kumar, Helleseth and Calderbank (IT-March 95) showed that

$$\sum_{x\in\mathcal{T}_{ms}} i^{T_1^{ms}(x)} = -\eta^s$$

where η is a complex number of magnitude $\sqrt{2^m}$. Further, since

$$\begin{aligned} \sum_{x\in\mathcal{T}_m} i^{T_1^m(x)} &= -(-\sum_{x\in\mathcal{T}_1} i^{T_1^1(x)})^m \\ &= (1+i)^m. \end{aligned}$$

Using (1), one can easily obtain the complete Lee weight distribution of the Kerdock code. (A similar idea was used in '70s by McEliece, Rumsey and Baumert to study irreducible binary codes).

III. Generalized Hamming Weights (GHW) of Z_4-Linear Codes

The rth generalized Hamming weight (minimum support weight) d_r of a Z_4-linear code $\mathcal{C}$ is the minimum size of the support of a subcode of $\mathcal{C}$ of size 4^r, $r = 0.5, 1, 1.5, \cdots$ (see Ashikhmin [Trondheim ISIT Proc. 1994]). Further, in (Ashikhmin [Trondheim ISIT Proc. 1994]), $d_{0.5}, d_1$ for the Kerdock code are determined and lower bounds on $d_{1.5}, d_2$ are obtained.

We will now show how we can sometimes compute the GHW for the Kerdock code of length 2^{ms} from the Kerdock code of length 2^m. For simplicity, in what follows, we assume the constant term in the trace expansion of the Kerdock code to be zero. For the Z_4 Kerdock code of length 2^m, it can be shown that

$$\begin{aligned} 4^r d_r &= 2^m - max(\Re\{\sum_{c\in B}\sum_{x\in\mathcal{T}_m} i^{c(x)}\}) \\ &= 2^m - max(\Re\{\sum_{c\in B}\Gamma_m(a_c+2b_c)\}) \end{aligned}$$

where maximum is over all submodules B of $\mathcal{C}$ of size 4^r. Similarly for the Z_4 Kerdock code of length 2^{ms},

$$4^r d_r = 2^{ms} - max(\Re\{\sum_{c\in B}\Gamma_{ms}(a_c+2b_c)\})$$

where again maximum is over all submodules B of $\mathcal{C}$ of size 4^r. If $a_c + 2b_c \in GR(4, m)$, then from earlier results,

$$\Gamma_{ms}(a_c + 2b_c) = (\Gamma_m(a_c + 2b_c))^s, \; s \text{ odd}.$$

For appropriate values of s, we note that good submodules of length 2^m can be lifted to good submodules of length 2^{ms}.

These results can also be extended to Z_4 Delsarte-Goethals' code, duals of double error correcting BCH codes, etc.

Improved Bezout Theorem and Its Applications in Coding Theory

Gui-Liang Feng, T. R. N. Rao, and Gene A. Berg‡*

Abstract

This paper presents an improved Bezout theorem which can be used to determine a tighter lower bound of the number of distinct points of intersection of two or more curves for a large class of plane curves. A new approach to determine a lower bound on the minimum distance for algebraic-geometric codes defined from a class of plane curves is introduced, based on the improved Bezout theorem. Examples of more efficient linear codes are constructed using the improved Bezout theorem and the new approach. For $d = 4$, the linear codes constructed by the new construction are better than or equal to the known linear codes. For $d \geq 5$, these new codes are better than the known codes. The Klein code (22, 15, 6) over $GF(2^3)$ is also constructed.

Index Terms: algebraic-geometric codes, linear codes, minimum distance, generalized Hamming weights, Bezout's theorem.

* Gui-Liang Feng, T. R. N. Rao are with the Center for Advanced Computer Studies, University of Southwestern Louisiana, Lafayette, LA. 70504, USA. email: glf@swamp.cacs.usl.edu and rao@swamp.cacs.usl.edu. This work was supported in part by the National Science Foundation under Grant NCR-9305038 and Louisiana Education Quality Support Fund under Grant LEQSF-(1994-96)-RD-A-36.

‡ Gene A. Berg is with the National Security Agency, Fort George G. Meade, MD 20755-6000,
email: geneberg@romulus.ncsc.mil.

I. Introduction

The minimum distance is one of most important parameters in error-correcting codes. It is a measure of the code's capacity to correct errors or detect errors or both [1]. The minimum distance d of a linear code C is defined by

$$d = \min_{\mathbf{u}, \mathbf{v} \in C} \{ d(\mathbf{u}, \mathbf{v}) \},$$

where $d(\mathbf{u}, \mathbf{v})$ expresses the Hamming distance between $\mathbf{u}$ and $\mathbf{v}$.

From this definition, we have the following lemma.

Lemma 1.1: Let C be an (n, k) linear code and $\mathbf{H}$ be a parity check matrix. If any $(d^* - 1)$ columns of $\mathbf{H}$ have rank $(d^* - 1)$, then the minimum distance d is at least d^*, i. e., $d \geq d^*$.

Let *LS* be a set of distinct points in a plane or a set of distinct roots of a plane curve (i.e. a polynomial). Let $LS = \{ (x_1, y_1), (x_2, y_2), ..., (x_n, y_n) \}$ and let $h(x,y)$ be a polynomial (or monomial), then a vector $(h(x_1, y_1), h(x_2, y_2), ..., h(x_n, y_n))$ is called an evaluated vector of polynomial $h(x,y)$ on the set *LS*. Hereinafter, sometimes $\mathbf{h}_i$ expresses an evaluated vector and sometimes it expresses a polynomial or a curve if no confusion arises. Thus, let $D\{ \mathbf{h}_{v_1}, \cdots, \mathbf{h}_{v_p} \}$ denote an upper bound on the number of distinct points of intersection of curves $\mathbf{h}^*_{v_1} = 0, ..., \mathbf{h}^*_{v_p} = 0$ for the case of *LS* being the set of all points in a whole plane, or denote an upper bound on the number of distinct points of the intersection of curves $\mathbf{h}^*_{v_1} = 0, ..., \mathbf{h}^*_{v_p} = 0$, and $f(x,y) = 0$ for the case of *LS* being the set of all points on the curve $f(x,y) = 0$, where $\mathbf{h}^*_{v_\mu}$ is a linear combination of $\mathbf{h}_{v_\mu}$ and its previous curves. Let $D^{(r)}_p$ for a given sequence of curves express the maximal possible number of distinct points of intersection of p curves among the first r curves of the given sequence of curves. Therefore, the calculation of $D^{(r)}_p$ reduces to the calculation of the number of distinct points of intersection of several curves. The Bezout theorem is a most well-known theorem that can be applied to solving the problem of determining the number of points of intersection of two curves.

Using these new notations, Lemma 1.1 can be transformed to the following lemma>

Lemma 1.2: Let C be an (n, k) linear code and $\mathbf{H}$ be a parity check matrix. If $D^{(r)}_{r-d^*+2} < d^* - 1$, where $r = n - k$, then the minimum distance d is at least d^*, i. e., $d \geq d^*$.

The organization of this paper is as follows. In the next section, we present an improved Bezout theorem, which can be used to determine an upper bound of the number of points of intersection of several plane curves. We also derive some properties of $D\{ \mathbf{h}_{v_1}, \cdots, \mathbf{h}_{v_p} \}$ and $D^{(r)}_p$. In Section III, we introduce a new construction, by which many more efficient linear codes (include the algebraic-geometric codes) can be easily constructed. Some conclusions are included in Section IV.

II. Improved Bezout Theorem

In this section, we first introduce Bezout's theorem. Then we derive an improved Bezout theorem and some properties of $D\{ \mathbf{h}_{v_1}, \cdots, \mathbf{h}_{v_p} \}$ and $D^{(r)}_p$, which are useful for the next section.

Bezout Theorem:([2]) Let $F(X,Y,Z)$ and $G(X,Y,Z)$ be two plane curves without common components, i.e., F and G are homogeneous forms of degree n and m, respectively, with no common factors. Then the intersection of $F(X,Y,Z)$ and $G(X,Y,Z)$ has at most mn distinct common points (X,Y,Z).

Definition 2.1: The x-resultant matrix, denoted by $RM(f,g)$ (or RM) of two polynomials

$$f(x) = a_0 x^n + a_1 x^{n-1} + \cdots + a_n$$
$$g(x) = b_0 x^m + b_1 x^{m-1} + \cdots + b_m$$

is given by the following $(m+n) \times (m+n)$ matrix:

$$\begin{bmatrix}
a_0 & a_1 & \cdots & \cdots & \cdots & a_n & & & & \\
 & a_0 & a_1 & \cdots & \cdots & \cdots & a_n & & & \\
 & & \cdot & \cdot & \cdot & \cdot & \cdot & & & \\
 & & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & & \\
 & & & & a_0 & a_1 & \cdots & \cdots & \cdots & a_n \\
b_0 & b_1 & \cdots & \cdots & b_m & & & & & \\
 & b_0 & b_1 & \cdots & \cdots & b_m & & & & \\
 & & \cdot & \cdot & \cdot & \cdot & & & & \\
 & & \cdot & \cdot & \cdot & \cdot & & & & \\
 & & & & & & b_0 & b_1 & \cdots & b_m
\end{bmatrix}$$

and its determinant is called the x-resultant of the two polynomials and denoted by $Res_x(f,g)$ (or R).

Theorem 2.1: $f(x)$ and $g(x)$ have a greatest common divisor polynomial with degree r if and only if rank($RM(f,g)$) $= m + n - r$.

For convenience in the following discussion, we define

$$\vec{f}^{(0)} \triangleq \vec{f} \triangleq (a_0, a_1, \cdots, a_n, 0, \cdots, 0),$$

where on the rightmost side there are $m - 1$ 0's, and

$$\vec{f}^{(i)} \triangleq (0, \cdots, 0, a_0, a_1, \cdots, a_n, 0, \cdots, 0),$$

where on the leftmost side there are i 0's ($0 \leq i \leq m - 1$) and on the rightmost side there are $m - i - 1$ 0's. Thus, the above matrix consists of the vectors $\vec{f}^{(\mu)}$ and $\vec{g}^{(\lambda)}$, for $0 \leq \mu \leq m - 1$ and $0 \leq \lambda \leq n - 1$.

Sometimes the x-resultant is called the Sylvester resultant because it was introduced by Sylvester [3]. In his paper, Sylvester showed that $Res_x(f,g) = 0$ if and only if either $a_0 = b_0 = 0$ or if f and g have a common root.

The coefficients of f and g could be polynomials in y. We could have:

$$f(x,y) = a_0(y)x^m + a_1(y)x^{m-1} + \cdots + a_m(y)$$
$$g(x,y) = b_0(y)x^n + b_1(y)x^{n-1} + \cdots + b_n(y)$$

Then for $R(y) = Res_x(f,g)$ and for any value β of y we would have $R(\beta) = 0$ if and only if either $a_0(\beta) = b_0(\beta) = 0$ or $f(\alpha,\beta) = g(\alpha,\beta) = 0$ for some α. Thus, the roots of $R(y) = 0$ are the projections of the points of intersection of f and g. In fact, the resultant gives more precise information. Namely, if the order of the zero β of $R(y)$ is e, i.e., if $R(y) = (y-\beta)^e D(y)$ with $D(\beta) \neq 0$, then counting properly, there are exactly e points of intersection of $f(x,y)$ and $g(x,y)$ lying on $y = \beta$. If it is not counted properly, there are at most e distinct points of intersection of $f(x,y)$ and $g(x,y)$ lying on $y = \beta$. Therefore, we have the following theorem.

Theorem 2.2: The number of the distinct points of intersection of two polynomials $f(x,y)$ and $g(x,y)$ without common components is at most equal to the degree of their resultant $R(y)$.

Proof: If β is a root of $R(y) = 0$ and its order is equal to r, then the $x-resultant$ matrix $RM(\beta)$ has rank $n + m - r$. From Theorem 2.1, there are at most r distinct values (denoted by α) of x such that (α,β) are the points in the intersection of $f(x,y) = 0$ and $g(x,y) = 0$, i.e., there are most r common roots with $y = \beta$ of $f(x,y) = 0$ and $g(x,y) = 0$. Thus, for each root β with an order r of $R(y) = 0$, there are at most r distinct points (α, β) in the intersection of $f(x,y) = 0$ and $g(x,y) = 0$. On the other hand, the summation of all root orders of equation $R(y) = 0$ is equal to the degree of $R(y)$. Therefore, the number of distinct points in the intersection of $f(x,y) = 0$ and $g(x,y) = 0$ is at most the degree of $R(y)$. The proof of Theorem 2.2 is completed. □

From Theorem 2.2, the Bezout theorem can be derived. If $f(x,y)$ and $g(x,y)$ have no common components, then their resultant $R(y)$ is not identical with zero. Thus, $\deg R(y) > -\infty$, we define $\deg 0 = -\infty$.

Now we will generalize Theorem 2.1. Let us consider p curves in affine plane curves without common components, i.e., $f_\mu(x,y) = 0$ for $\mu = 1, 2, ..., p$. Without loss of the generality, $deg_x\, f_1 \geq deg_x\, f_2 \geq \cdots \geq deg_x\, f_p$, and let $deg_x\, f_1 = m$ and $deg_x\, f_2 = n$, were $deg_x\, f_\mu$ indicates the maximal i such that the monomial $x^i y^j$ is a term in f_μ. We define the x-resultant matrix of these p curves or polynomials as the following $\Sigma \times (m+n)$ matrix, where $\Sigma = \sum_{\mu=1}^{p} (m+n-deg_x\, f_\mu)$ and $s = deg_x\, f_p$:

$$\begin{bmatrix}
a_0^{(1)} & a_1^{(1)} & \cdot & \cdot & \cdot & a_m^{(1)} & 0 & \cdot & \cdot & 0 \\
0 & a_0^{(1)} & a_1^{(1)} & \cdot & \cdot & \cdot & a_m^{(1)} & 0 & \cdot & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
0 & 0 & \cdot & 0 & a_0^{(1)} & a_1^{(1)} & \cdot & \cdot & \cdot & a_m^{(1)} \\
a_0^{(2)} & a_1^{(2)} & \cdot & \cdot & a_n^{(2)} & 0 & 0 & \cdot & \cdot & 0 \\
0 & a_0^{(2)} & a_1^{(2)} & \cdot & \cdot & a_n^{(2)} & 0 & \cdot & \cdot & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
0 & 0 & \cdot & \cdot & \cdot & 0 & a_0^{(2)} & a_1^{(2)} & \cdot & a_n^{(2)} \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
a_0^{(p)} & a_1^{(p)} & \cdot & \cdot & a_s^{(p)} & 0 & 0 & \cdot & \cdot & 0 \\
0 & a_0^{(p)} & a_1^{(p)} & \cdot & \cdot & a_s^{(p)} & 0 & \cdot & \cdot & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
0 & 0 & \cdot & \cdot & \cdot & 0 & a_0^{(p)} & a_1^{(p)} & \cdot & a_s^{(p)}
\end{bmatrix}$$

Let $R(y) = Res_x(f_1, f_2, \cdots, f_p)$ be the non-zero determinant of nonsingular submatrix with the smallest degree of y of the x-resultant matrix. Similar to the proof of Theorem 2.2, we have the general theorem as follows.

Theorem 2.3: The number of distinct points of the intersection of $f_\mu(x,y)$ without common components, for $\mu = 1, 2, \ldots, p$, is at most equal to the degree of their resultant $R(y)$, i.e., $deg\ R(y)$.

In order to get an upper bound of $deg\ R(y)$, we introduce a new concept. Among the f's with the same degree of y, we select one. Thus, we can select f_{λ_μ}, for $\mu = 1, 2, \ldots, q\ (\le p)$, such that $deg_x f_{\lambda_i} > deg_x f_{\lambda_{i+1}}$, $\{ deg_x f_{\lambda_\sigma} \mid \sigma = 1, 2, \ldots, q \} = \{ deg_x f_\mu \mid \mu = 1, 2, \ldots, p \}$, and f_{λ_σ} have no common components. We define the x-partial resultant matrix of these p curves or polynomials as the following $(m+n)\times(m+n)$ matrix:

$$[\vec{f}_{\lambda_1}^{(0)}, \ldots, \vec{f}_{\lambda_1}^{(d_2-1)}, \vec{f}_{\lambda_2}^{(d_2)}, \ldots, \vec{f}_{\lambda_2}^{(d_1-1)}, \ldots, \vec{f}_{\lambda_q}^{(d_1+d_2-d_{q-1})}, \ldots, \vec{f}_{\lambda_q}^{(d_1+d_2-d_q-1)}]^T,$$

where d_σ denotes $deg_x\ f_{\lambda_\sigma}$.

Obviously, this matrix is an upper triangle matrix when $d_q = 0$. The determinant of this matrix can be easily calculated for the special case, i.e., the determinant is equal to the product of all elements on main diagonal of this matrix. This determinant is called a partial resultant and denoted by $PR(y)$.

Corollary 2.1: The number of distinct points of the intersection of $f_\mu(x,y)$, for $\mu = 1, 2, \ldots, p$, is at most equal to the degree of their partial resultant $PR(y)$

Example 2.1: Let us consider the number of common points on the following four curves over $GF(2^4)$:

$$\begin{cases} x^5 + y^4 + y = 0 \\ x^3 + a(y)x^2 + b(y)x + c(y) = 0 \\ xy + e(y) = 0 \\ y^2 + fy + g = 0 \end{cases}$$

We have the following matrix:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & y^4+y & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & y^4+y & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & y^4+y \\ 0 & 0 & 0 & 1 & a(y) & b(y) & c(y) & 0 \\ 0 & 0 & 0 & 0 & 1 & a(y) & b(y) & c(y) \\ 0 & 0 & 0 & 0 & 0 & y & e(y) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & y & e(y) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & y^2+fy+g \end{bmatrix}$$

Thus, $PR(y) = y^2(y^2 + fy + g)$. Obviously, $deg\ PR(y) = 4$. Therefore, the number of distinct points of the intersection of the four curves is at most 4.

Remark (1): Here we regard $f_\mu(x,y)$ as a polynomial of x and the coefficients are polynomials in y. We also can regard $f_\mu(x,y)$ as a polynomial of y and the coefficients are polynomials in x. The number of the distinct points of intersection of $f_\mu(x,y)$'s is the same. The distinct points of intersection of $f_\mu(x,y)$'s obtained by the two approachs are also the same.

Remark (2): It is sufficient and necessary that f_μ, for $\mu = 1, 2, \ldots, p$, have no common components.

Definition 2.2: $D\{f_1, f_2, \cdots, f_p\}$ denote the number of distinct points of the intersection of curves $f_\mu^*(x,y) = 0$, for $\mu = 1, 2, \ldots, p$, where $f_\mu^*(x,y) = 0$ is a linear combination of $f_\mu(x,y) = 0$ and its previous curves.

Definition 2.3: Given a sequence of polynomials $\{ f_\mu(x,y) \mid \mu = 1, 2, \ldots, r \}$.

$$D_p^{(r)} = \max \{ D\{f_{v_1}, f_{v_2}, \cdots, f_{v_p}\} \mid v_1, \cdots, v_p \le r \},$$

where $f_{v_\mu}^*$ expresses a linear combination of f_i for $i = 1, 2, \ldots, \lambda_\mu$, and the coefficient of f_{λ_μ} is 1, i.e., $f_{\lambda_\mu}^* =$

$f_{\lambda_\mu} + \sum_{i=1}^{\lambda_\mu - 1} c_i f_i.$

We have the following results:

Proposition 2.1: $D\{ f(x,y) g(x,y) \} \leq D\{ f(x,y) \} + D\{ g(x,y) \}.$

Proposition 2.2: $D\{ f_1, \cdots, f_p \} \leq \min \{ D\{ f_\mu \} \mid \mu = 1, 2, \ldots, p \}$

From Proposition 2.1 and Proposition 2.2, we have:

Proposition 2.3: $D\{ gf_1, \cdots, gf_p \} \leq D\{ g \} + D\{ f_1, \cdots, f_p \}.$

Proposition 2.4: $D\{ gf_1, f_1, \cdots, f_p \} = D\{ f_1, \cdots, f_p \}$

Proposition 2.5: $D_p^{(r)} \geq D_{p+1}^{(r)} + 1.$

III. Construction of Efficient Linear Codes with Small Minimum Distance

In this section, three examples illustrate how the improved Bezout theorem can be used to construct more efficient linear codes including the algebraic geometric codes from plane curves.

III.1 Efficient Linear Codes with Minimum Distance 4

Let us consider $D_2^{(4)}$ over $GF(2^m)$, where the first four polynomials are $\{ 1, x, y, x^2 + \beta xy + y^2 \}$, where β is any element with $\mathrm{tr}(\beta^{-1}) \triangleq \sum_{i=0}^{m-1} \beta^{-2^i} = 1$ over $GF(2^m)$.

In the following, we will prove an important result:

Theorem 3.1: $D_2^{(4)} \leq 2.$

Proof: Since the four polynomials are $1, x, y, x^2 + \beta xy + y^2$, we have the following selections: $\{ 1, x \}$, $\{ 1, y \}$, $\{ 1, x^2 + \beta xy + y^2 \}$; $\{ x, y \}$, $\{ x, x^2 + \beta xy + y^2 \}$, and $\{ y, x^2 + \beta xy + y^2 \}$. From $D\{ 1 \} = 0$ and Lemma 2.3, $D\{ 1, x \} = D\{ 1, y \} = D\{ 1, x^2 + \beta xy + y^2 \} = 0$. Also, $D\{ x, y \} = 1$. Now we calculate the fifth selection, i.e. $D\{ x, x^2 + \beta xy + y^2 \}$. For any constants b, c, d, e, we calculate the number of distinct points of intersection of the following curves:

$$\begin{cases} x + b = 0 \\ x^2 + \beta xy + y^2 + cy + dx + e = 0 \end{cases}$$

We have

$$R(x) = \begin{vmatrix} 1 & \beta y + d & y^2 + cy + e \\ 1 & b & 0 \\ 0 & 1 & b \end{vmatrix}$$

i.e., $R(y) = y^2 + cy + e + \cdots$. Thus, $deg\ R(y) = 2$. From Theorem 2.2, $D\{ x, x^2 + \beta xy + y^2 \} \leq 2$.

Now let us consider the last selection. For any constants a, b, c, d, e, we calculate the number of distinct points of intersection of the following curves:

$$\begin{cases} y + ax + b = 0 \\ x^2 + \beta xy + y^2 + cy + dx + e = 0 \end{cases}$$

We have

$$R(x) = \begin{vmatrix} 1 & \beta x + c & x^2 + dx + e \\ 1 & ax + b & 0 \\ 0 & 1 & ax + b \end{vmatrix}$$

i.e., $R(x) = a^2x^2 + b^2 + x^2 + dx + e - \beta ax^2 - acx - \beta bx - bc = (a^2 + \beta a + 1)x^2 + (d + \beta b + ac)x + bc + b^2$. Since $\mathrm{tr}(\beta^{-1}) = 1$, $a^2 + \beta a + 1 \neq 0$ for all $a \in GF(2^m)$. Thus, deg $R(x) = 2$. Therefore, $D_{\{y,\, x^2+\beta xy+y^2\}} \leq 2$.

Combining the above results, we have $D_2^{(4)} \leq 2$. □

Remark (3): For the maximal number of distinct points of intersection, the following forms are equivalent:

$$\begin{cases} y + ax + b = 0 \\ x^2 + \beta xy + y^2 + cy + dx + e = 0 \end{cases}$$

and

$$\begin{cases} y + ax + b = 0 \\ x^2 + \beta xy + y^2 + dx + e = 0 \end{cases}$$

Hereinafter, we use the second form for simplification.

Remark (4): Observe that $D_{\{\mathbf{h}_{v_1}, \ldots, \mathbf{h}_{v_p}\}} \leq l$, if and only if $-\infty <$ deg of $R(x) \leq l$. Further, if the degree of $R(x)$ is equal to $-\infty$, i.e., $R(x)$ is a zero, then $R(x) = 0$ has all x for solutions.

Theorem 3.2: Let C be a linear code with length $n = 2^{2m}$ over $GF(2^m)$, which is defined by a parity check matrix $\mathbf{H} \triangleq [\,1, x, y, x^2 + \beta xy + y^2\,]^T$. Then the minimum distance of C is at least 4.

Proof: From Theorem 3.1, $D_2^{(4)} = D_{\{y,\, x^2+\beta xy+y^2\}} \leq 2 < 4\text{-}1$. From Lemma 1.2, the minimum distance of C is at least 4. □

Example 3.1: Let $m = 2$, and let α be a primitive element of $GF(2^2)$, $\beta = \alpha^{-1}$. It can be easily checked that $\alpha + \alpha^2 = 1 \neq 0$. From Theorem 3.2, the following matrix defines a linear codes C with length 16 and minimum distance at least 4 over $GF(2^2)$:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \alpha & \alpha & \alpha & \alpha & \alpha^2 & \alpha^2 & \alpha^2 & \alpha^2 \\ 0 & 1 & \alpha & \alpha^2 & 0 & 1 & \alpha & \alpha^2 & 0 & 1 & \alpha & \alpha^2 & 0 & 1 & \alpha & \alpha^2 \\ 0 & 1 & \alpha^2 & \alpha & 1 & \alpha^2 & \alpha^2 & 1 & \alpha^2 & \alpha^2 & \alpha & \alpha & \alpha & 1 & \alpha & 1 \end{bmatrix}$$

Construction 3.1: Let $n = 2^{2km}$, The following parity check matrix $\mathbf{H}$ defines a linear code C with length $n = 2^{2km}$ and minimum distance at least 4 over $GF(2^m)$, i.e., C is a $(2^{2km}, 2^{2km} - 1 - 3k, \geq 4)$ code:

$$\mathbf{H} = [1\,, x_1\,, y_1\,, x_1^2 + \beta x_1 y_1 + y_1^2\,, \cdots, x_k\,, y_k\,, x_k^2 + \beta x_k y_k + y_k^2\,]^T\,,$$

where $\mathrm{tr}(\beta^{-1}) = 1$.

These codes are better than the Kaneda codes [4], and same as the Chen codes [5]. However, the derivations are different. The codes obtained by Construction 4.1 can be more easily decoded than the Chen codes.

III.2 Efficient Linear Codes with Minimum Distance ≥ 5

Let us consider $D_3^{(7)}$ over $GF(2^m)$, where the polynomials are $\{1, x, y, x^2, xy, y^2, x^3 + \gamma x^2 y + \beta xy^2 + y^3\}$, where $\gamma, \beta \in GF(2^m)$ and $x^3 + \gamma x^2 y + \beta xy^2 + y^3$ is irreducible. We have the following theorem.

Theorem 3.3: $D_4^{(7)} \leq 3$.

Proof: Let $D_4^{(7)} = D_{\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}}$. If $\lambda_1 = 1$, then from Proposition 2, $D_{\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}} \leq D_{\{1\}} = 0$.

If $\lambda_1 = 2$ or 3, then it can be easily checked that $D\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\} \leq 3$. Now we need only to consider the case of $\lambda_1 = 4$, i.e., $D\{x^2, xy, y^2, x^3 + \gamma x^2 y + \beta xy^2 + y^3\}$. Let us consider the following curves:

$$\begin{cases} x^2 + ay + bx + c = 0 \\ xy + dy + ex + f = 0 \\ y^2 + gy + hx + i = 0 \\ x^3 + \gamma x^2 y + \beta xy^2 + y^3 + jy + kx + l = 0 \end{cases}$$

where $a, b, \cdots, l$ are any constants of $GF(2^m)$. We have the following partial x-resultant matrix:

$$\begin{vmatrix} 1 & \gamma y & \beta y^2 + k & y^3 + jy + l & 0 \\ 0 & 1 & \gamma y & \beta y^2 + k & y^3 + jy + l \\ 0 & 0 & 1 & b & ay + c \\ 0 & 0 & 0 & y + e & dy + f \\ 0 & 0 & 0 & h & y^2 + gy + i \end{vmatrix}$$

$PR(y) = (y + e)(y^2 + gy + i) - h(dy + f)$. $degPR(y) = 3$.

Combining the above results, the proof is completed. □

From Lemma 1.2 and Theorem 3.3, we have the following theorem.

Theorem 3.4: Let C be a linear code with length $n = 2^{2m}$ over $GF(2^m)$, which is defined by a parity check matrix $\mathbf{H} \triangleq [1, x, y, x^2, xy, y^2, x^3 + \gamma x^2 y + \beta xy^2 + y^3]^T$, where $x^3 + \gamma x^2 y + \beta xy^2 + y^3$ is a irreducible polynomial over $GF(2^m)$. Then the minimum distance of C is at least 5.

Example 3.2: Let $m = 2$, and let α be a primitive element of $GF(2^2)$. It can checked that $x^3 + \alpha x^2 y + y^3$ is an irreducible polynomial. From Theorem 3.3.2, the following matrix defines a linear code C (16, 9, ≥5) over $GF(2^2)$:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \alpha & \alpha & \alpha & \alpha & \alpha^2 & \alpha^2 & \alpha^2 & \alpha^2 \\ 0 & 1 & \alpha & \alpha^2 & 0 & 1 & \alpha & \alpha^2 & 0 & 1 & \alpha & \alpha^2 & 0 & 1 & \alpha & \alpha^2 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \alpha^2 & \alpha^2 & \alpha^2 & \alpha^2 & \alpha & \alpha & \alpha & \alpha \\ 0 & 0 & 0 & 0 & 0 & 1 & \alpha & \alpha^2 & 0 & \alpha & \alpha^2 & 1 & 0 & \alpha^2 & 1 & \alpha \\ 0 & 1 & \alpha^2 & \alpha & 0 & 1 & \alpha^2 & \alpha & 0 & 1 & \alpha^2 & \alpha & 0 & 1 & \alpha^2 & \alpha \\ 0 & 1 & 1 & 1 & 1 & \alpha & \alpha^2 & 1 & 1 & 1 & \alpha & \alpha^2 & 1 & \alpha^2 & 1 & \alpha \end{bmatrix}$$

Remark (5): The result can be generalized to the cases of any minimum distance $d \geq 6$. The code dimension can be increased by at least one, while keeping the same code length $n = 2^{2m}$ and the minimum distance $d \geq 6$ over $GF(2^m)$ [6, 7].

III.3. More Efficient Algebraic-Geometric Codes

Let us consider the Klein curve over $GF(2^3)$:

$$x^3 y + y^3 + x = 0. \tag{3.1}$$

There are 22 points on the curve. Let LS be the set of all points on the curve. The LS has the following points:

$x =$	$y =$	$x =$	$y =$	$x =$	$y =$
0	0;				
1	$\alpha, \alpha^2, \alpha^4$;				
α	$1, \alpha^2, \alpha^6$;	α^2	$1, \alpha^4, \alpha^5$;	α^4	$1, \alpha, \alpha^3$;
α^3	$\alpha^2, \alpha^3, \alpha^5$;	α^6	$\alpha^4, \alpha^6, \alpha^3$;	α^5	$\alpha, \alpha^5, \alpha^6$,

where α is a primitive element of $GF(2^3)$.

Let us consider a linear code C defined by the parity check matrix

$$\mathbf{H} \triangleq [\,1, x, y, x^2, xy, x^3\ , y^2\,]^T$$

Theorem 3.5: The minimum distance of the code C is at least 6, i.e., C is a linear code (22, 15, ≥6) over $GF(2^3)$.

Proof. From Lemma 1.2, it is sufficient to prove $D_3^{(7)} \le 4$. From the definition,
$D_3^{(7)} = \max \{ D_{\{1,*,*\}}, D_{\{x,*,*\}}, D_{\{y,*,*\}}, D_{\{x^2, xy, x^3\}}, D_{\{x^2, xy, y^2\}}, D_{\{x^2, x^3, y^2\}}, D_{\{xy, x^3, y^2\}} \}$.

It is easily seen (using Propositions 1-4) that the first member is equal to zero, the second, third, and 5-th members are at most 3, and the 4-th and 6-th members are at most 4. In the following we prove that the 7-th member is at most 4.

From the definition, $D_{\{xy, x^3, y^2\}}$ expresses the maximal number of distinct points of intersection of the following four curves:

$$\begin{cases} x^3y + y^3 + x = 0 \\ xy + ax^2 + by + cx + d = 0 \\ x^3 + ex^2 + fy + gx + h = 0' \\ y^2 + ix^2 + jy + kx + l = 0 \end{cases} \tag{3.2}$$

where $a, b, c, d, e, f, g, h, i, j, k, l$ are any constants over $GF(2^3)$. From these equations, we have the following $PR(y)$:

$$\begin{vmatrix} 1 & e & g & fy+h \\ a & y+c & by+d & 0 \\ 0 & a & y+c & by+d \\ 0 & i & k & y^2+jy+l \end{vmatrix}$$

It can be easily checked that the degree of the determinant is 4, i.e., the degree of $PR(y)$ is at most 4. Thus, the number of distinct points of intersection of these four curves is at most equal to 4. Hence, $D_{\{xy, x^3, y^2\}} \le 4$. Combining the other members, we have $D_3^{(7)} \le 4$. □

Using the result in [6], we can only obtain a linear code (22, 14, ≥6) over $GF(2^3)$.

Let us consider another example. Let C_6 be a modifined Klein code defined by the following parity check matrix:

$$\mathbf{H}_6 = [\ 1\ , x\ , y\ , x^2\ , xy\ , x^3 + y^2\ ]^T .$$

Using the improved Bezout theorem, it can be proved that $D_3^{(6)} \le 3$. Thus, the minimum distance of C_6 is at least 5. In fact, the true minimum distance is 5. Therefore, C_6 is a (22,16,5) code over $GF(2^3)$. This code is better than the current Klein code and the improved Klein code with 6 check bits.

IV. Conclusions

It is well known that the Bezout theorem can be used to determine an upper bound of the number of common points of two plane curves. In this paper, we first reduce the determination of a lower bound of a linear code's minimum distance to the determination of the number of distinct points of intersection of several curves. Then, we present an improved Bezout theorem and a new approach to determine the minimum distance We also present a new construction of linear codes with any length n over GF(2^b) with minimum distance 4 and ≥ 5. The codes with $d = 4$ have been used in computer memory systems. The new codes have applications not only in computer memory systems but also in distributed systems [8, 9], CD audio, Video disk, and CD ROM.

In this paper, we discuss only the case in two-dimensional affine spaces. Our results should be generalized to high-dimensional affine spaces.

References

[1] R. E. Blahut, *Theory and Practice of Error Control Codes*, Reading, Ma: Addison-Wesley, 1985.

[2] S. S. Abhyankar, *Algebraic Geometry for Scientists and Engineers*, The American Mathematical Society, 1990.

[3] J.J. Sylvester, "On a general method of determining by mere inspection the derivations from two equations of any degree", *Philosophical Magazine* **16**(1840), 132-135.

[4] S. Kaneda and E. Fujiwara, "Single byte error correcting-double byte error detecting codes for memory systems," *IEEE Trans. on Computers* Vio. C-31, pp.596-602, July, 1982.

[5] C. L. Chen, "Byte-oriented error-correcting codes for semiconductor memory systems," *IEEE Trans. on Computers* Vio. C-35, pp.646-648, July, 1986.

[6] G. L. Feng and T. R. N. Rao, "Improved Geometric Goppa Codes, Part I: Basic Theory" to appear in *IEEE Trans. on Information Theory.*

[7] C. Heegard and K. Saints, "Cascaded Reed-Solomon codes and Grobner Bases," *Proceedings of the 1993 IEEE Workshop on Information Theory* pp.5.2.1-5.2.2, June 4-8, 1993, Japan.

[8] M. O. Rabin, "Efficient dispersal of information for security, load balancing and fault-tolerance," Harvard University, Cambridge, MA. TR-02-87, Apr. 1987. G. Agrawal and P. Jalote, "Coding-based replication schemes for distributed systems," *IEEE Trans. on Parallel and Distributed Systems* Vol. 6, pp. 240-251, Mar. 1995.

The Algebraic Structure of Some AG Goppa Codes

John Little
Department of Mathematics
College of the Holy Cross
Worcester, MA 01610
email: `little@math.holycross.edu`

September 29, 1995

Abstract

The curves used to construct many interesting AG Goppa codes have large groups of automorphisms, which induce automorphisms of the corresponding codes. As a result, many of these codes have useful *module* structures over the group rings of various subgroups of the automorphism group of the curve. In this paper the algebraic structure of these modules is examined using the theory of Groebner bases and applications to both encoding and decoding are presented.

§1. Preliminaries.

Let $C \subseteq \mathbf{F}_q^n$ be a linear code over $\mathbf{F}_q$ which has a non-trivial automorphism σ. The cyclic codes of block length $q-1$ are perhaps the most familiar examples of this type. Many interesting AG Goppa codes also have such automorphisms. Recall that if X is a smooth algebraic curve defined over the finite field $\mathbf{F}_q$, and $D = \sum_{i=1}^n P_i$ and G are $\mathbf{F}_q$-rational divisors on X with disjoint supports, then the Goppa code $C_L(D,G)$ is defined to be

$$C_L(D,G) = \{(f(P_1),\ldots,f(P_n)) \in \mathbf{F}_q^n \mid f \in L(G)\}$$

(where $L(G)$ is the vector space of rational functions on X defined over $\mathbf{F}_q$, whose divisor of zeroes and poles is bounded by G.)

(1.1) Example. Let X_m be the Hermitian curve over $\mathbf{F}_{m^2}$, defined by the equation

$$x^{m+1} = y^m + y\ .$$

Write Q for the single point at infinity of this curve, and let D be the sum of the other m^3 $\mathbf{F}_{m^2}$-rational points, each with coefficient 1. Writing α for a generator of $\mathbf{F}_{m^2}^*$, as in [1], consider the automorphism

$$(1) \qquad \sigma : \begin{cases} x \mapsto \alpha x \\ y \mapsto \alpha^{m+1} y \end{cases}$$

of X_m. Note that σ has order m^2-1 in the automorphism group $Aut(X_m)$. Since σ fixes the divisors D and $G = aQ$, σ induces an automorphism of each of the codes $C_L(D, aQ)$ constructed from X_m.

(1.2) Example. Let $q = 2^{2n+1}$, $q_0 = 2^n$ and let Y_n be the curve over $\mathbf{F}_q$ defined by the equation

$$z^q + z = y^{q_0}(y^q + y)$$

These curves have been studied by Hansen and Stichtenoth in [2]. With $n = 1$, for example, the curve over $\mathbf{F}_8$ in this family:

$$z^8 + z = y^2(y^8 + y) \tag{2}$$

has degree 10, genus 14, and a single (singular) point Q at infinity. The singularity at Q is a "cuspidal" singularity however – more precisely, there is only one point $\widetilde{Q}$ that lies over Q on the normalization (smooth model) $\widetilde{Y_1} \to Y_1$. From the point of view of coding theory, Y_1 is interesting because it has as many $\mathbf{F}_8$-rational points as possible for a curve of genus 14. Indeed, it passes through *every* point of the affine plane over $\mathbf{F}_8$. One can use $G = a\widetilde{Q}$ and D the sum of the 64 $\mathbf{F}_8$-rational affine points, each with coefficient 1, in Goppa's construction. The mapping

$$\sigma : \begin{cases} y \mapsto \alpha y \\ z \mapsto \alpha^{q_0+1} z \end{cases} \tag{3}$$

restricts to an automorphism of Y_n. Since σ fixes the divisors D and $G = a\widetilde{Q}$, σ induces an automorphism of each of the codes $C_L(D, a\widetilde{Q})$ constructed from Y_n. σ has order $q-1$ in $Aut(Y_n)$.

To describe a general context containing these examples, consider codes with an automorphism σ of order $q-1$, under the action of the cyclic group generated by σ. For instance, a cyclic code of block length $q-1$ has such an automorphism and is an ideal I in the ring

$$R = \mathbf{F}_q[t]/\langle t^{q-1} - 1\rangle \ ,$$

(isomorphic to the group ring of the cyclic group generated by σ.)

In general, *e.g.* if the block length is greater than $q-1$, the entries of the codewords of a code with an automorphism of order $q-1$ will be cyclically permuted in several blocks by σ. Decompose the entries of words (or equivalently, the standard basis vectors $\mathbf{v}_i$ in $\mathbf{F}_q^n$) into disjoint $\langle\sigma\rangle$-orbits

$$O_1 \cup \cdots \cup O_r \ .$$

Picking any one element $\mathbf{v}_{i,0}$ from each orbit as representative, relabel the basis vectors in the following way

$$\mathbf{v}_{i,j} = \sigma^j(\mathbf{v}_{i,0}) \ ,$$

and relabel the entries of each word correspondingly. If $c \in C$, we will write $c_{i,j}$ = coefficient of $\mathbf{v}_{i,j}$ in c.

The codewords can be written as r-tuples of polynomials $(h_1(t), \ldots, h_r(t))$, where

$$h_i(t) = \sum_{j=0}^{|O_i|-1} c_{i,j} t^j \ , \tag{4}$$

and under this identification the code becomes a vector subspace C of

$$\bigoplus_{i=1}^{r} \mathbf{F}_q[t]/\langle t^{|O_i|} - 1\rangle \ .$$

As in [1], Proposition (2.1.4), because C is closed under the action of σ, the collections of r-tuples of polynomials are closed under multiplication by t. In other words, this construction gives C the structure of a *module* over the ring $\mathbf{F}_q[t]$, or over the group ring R. Furthermore, there is an obvious surjection

$$\mathbf{F}_q[t]^r \xrightarrow{\pi} \bigoplus_{i=1}^{r} \mathbf{F}_q[t]/\langle t^{|O_i|} - 1\rangle ,$$

so to each code C as above, one can associate the corresponding submodule $\overline{C} = \pi^{-1}(C)$ of the free module $\mathbf{F}_q[t]^r$.

To introduce Groebner bases, a *monomial ordering* in the free module $\mathbf{F}_q[t]^r$ must be chosen. A convenient one is the *position over term*, or *POT*, ordering. This is defined by firs ordering the standard basis vectors $\mathbf{e}_i$ as in

$$\mathbf{e}_1 > \mathbf{e}_2 > \cdots > \mathbf{e}_r .$$

Then

$$t^u \mathbf{e}_i >_{POT} t^v \mathbf{e}_j$$

if $i < j$, or if $i = j$ and $u > v$. There is a unique reduced Groebner basis $\mathcal{G}$ for any non-zero submodule $\overline{C} \subseteq \mathbf{F}_q[t]^r$ with respect to this ordering. By the properties of the *POT* ordering (see e.g. [1], (2.2.4)), $\mathcal{G} = \{g^{(1)}, \ldots, g^{(r)}\}$ will have the upper-triangular form:

$$\begin{aligned}
g^{(1)} &= (g_1^{(1)}, g_2^{(1)}, \ldots, g_r^{(1)}) \\
g^{(2)} &= (0, g_2^{(2)}, \ldots, g_r^{(2)}) \\
&\ \ \vdots \\
g^{(r)} &= (0, 0, \ldots, 0, g_r^{(r)})
\end{aligned}$$

(1.3) Example. Consider the code $C = C_L(D, 20Q)$ (parameters $n = 27$, $k = 18$, $d = 7$ over $\mathbf{F}_9$) from the Hermitian curve over $\mathbf{F}_9$, and the automorphism σ of order 8 as in (1) with $m = 3$. σ permutes the 27 affine $\mathbf{F}_9$-rational points in 5 orbits: three of length 8, one of length 2 and one of length 1. Using the normalization $\mathbf{F}_9 = \mathbf{F}_3[\alpha]/\langle \alpha^2 + \alpha - 1\rangle$ and choosing as orbit representatives the three affine points with $x = 1$, the point $(0, \alpha^2)$, and $(0, 0)$, the reduced *POT* Groebner basis $\mathcal{G}$ has the form:

$$\begin{aligned}
g^{(1)} &= (1, \alpha^6, \alpha^6 t^4 + \alpha^2 t^3 + \alpha^7 t^2 + \alpha^4 t + \alpha^6, \alpha^4, 1) \\
g^{(2)} &= (0, t + \alpha^5, \alpha^6 t^4 + t^3 + \alpha^4 t^2 + \alpha^4 t + \alpha^2, \alpha^7 t + \alpha^5, \alpha^6) \\
g^{(3)} &= (0, 0, t^5 + \alpha^7 t^4 + \alpha t^3 + t^2 + \alpha^6 t + \alpha^5, \alpha t + \alpha^6, \alpha^3) \\
g^{(4)} &= (0, 0, 0, t^2 - 1, 0) \\
g^{(5)} &= (0, 0, 0, 0, t - 1)
\end{aligned} \tag{5}$$

(1.4) Example. The code $C = C_L(D, 57\widetilde{Q})$ on the Hansen-Stichtenoth curve Y_1 has parameters $n = 64$, $k = 44$, $d \geq 7$. The automorphism σ from (3) permutes the points of D in 10 orbits, 9 of length 7 and one of length 1. The semigroup of pole orders at $\widetilde{Q}$ of rational functions on the (singular) curve Y_1 is generated by the natural numbers 8, 10, 12, 13 – $y \in L(8\widetilde{Q})$, $z \in L(10\widetilde{Q})$, $f = y^5 + z^4 \in L(12\widetilde{Q})$, and $g = yz^4 + y^{20} + z^{16} \in L(13\widetilde{Q})$ (see [2]). Using these functions to generate a basis for $L(57\widetilde{Q})$, the normalization $\mathbf{F}_8 = \mathbf{F}_2[\alpha]/\langle \alpha^3 + \alpha + 1\rangle$, orbit representatives $(1, \alpha^6), \ldots, (1, \alpha), (1, 1), (0, 1), (1, 0), (0, 0)$ (in that order) the reduced *POT* Groebner basis $\mathcal{G}$ has the form

$$
\begin{aligned}
g^{(1)} &= (1,0,0,0,0,*,*,*,*,*)\\
g^{(2)} &= (0,1,0,0,0,*,*,*,*,*)\\
g^{(3)} &= (0,0,1,0,0,*,*,*,*,*)\\
g^{(4)} &= (0,0,0,1,0,*,*,*,*,*)\\
g^{(5)} &= (0,0,0,0,1,*,*,*,*,*)\\
g^{(6)} &= (0,0,0,0,0,t^2+(\alpha^2+\alpha)t+\alpha+1,*,*,*,*)\\
g^{(7)} &= (0,0,0,0,0,0,t^4+(\alpha+1)t^3+(\alpha^2+1)t^2+\alpha^2+\alpha+1,*,*,*)\\
g^{(8)} &= (0,0,0,0,0,0,0,t^6+t^5+t^4+t^3+t^2+t+1,0,0)\\
g^{(9)} &= (0,0,0,0,0,0,0,0,t^7-1,0)\\
g^{(10)} &= (0,0,0,0,0,0,0,0,0,t-1)
\end{aligned}
\tag{6}
$$

(To save space, the coefficients in the non-leading terms are omitted. Also, it is only the leading terms that determine the information positions and parity check positions for the code – see §2 below.)

More generally, if H is *any* group of automorphisms (cyclic or not) of a code C, by essentially the same construction as above, C has the structure of a module over the group ring $\mathbf{F}_q[H]$.

(1.5) Example. The Hermitian curve X_3 has many automorphisms besides those in the subgroup generated by σ from (1). In fact, by [3], VII.4.6, there is a nonabelian subgroup H ($|H| = 8 \cdot 27 = 216$) that fixes both the point at infinity Q, and the divisor D, hence induces automorphisms of $C_L(D, aQ)$ for all a. The elements of this subgroup can be written as

$$
\tau_{\lambda,\delta,\mu} = \begin{cases} x \mapsto \lambda x + \delta \\ y \mapsto \lambda^4 y + \lambda\delta^3 x + \mu \end{cases}
\tag{7}
$$

where $\lambda \in \mathbf{F}_9^*$, and (δ, μ) is any affine $\mathbf{F}_9$-rational point on X_3. Note that

$$\tau_{\lambda,\delta,\mu}(0,0) = (\delta, \mu)$$

This implies that H acts *transitively* on the affine $\mathbf{F}_9$-rational points of X_3. Hence there is only one orbit of points under H, and the AG Goppa codes $C_L(D, aQ)$ are *ideals* in $\mathbf{F}_9[H]$.

(1.6) Example. Similarly, the Hansen-Stichtenoth curve Y_1 has many automorphisms besides those in the subgroup generated by σ from (3). There is a (non-abelian) subgroup H of $Aut(Y_1)$, of order 448 that fixes both $a\widetilde{Q}$ and D, hence induces automorphisms of each $C_L(D, a\widetilde{Q})$ code. The elements of this subgroup can be written as

$$
\tau_{\lambda,\delta,\mu} = \begin{cases} y \mapsto \lambda y + \delta \\ z \mapsto \lambda^3 z + \lambda\delta^2 y + \mu \end{cases}
\tag{8}
$$

where $\lambda \in \mathbf{F}_8^*$, and (δ, μ) is any affine $\mathbf{F}_8$-rational point on Y_1. Once again, H acts transitively on the points of D, and the codes $C_L(D, a\widetilde{Q})$ are *ideals* in the group algebra $\mathbf{F}_8[H]$.

§2. Application to Encoding.

The strong parallel with the cyclic case observed for a *cyclic* group of automorphisms as in (1.3) and (1.4) above suggests a systematic encoding method for a large class of AG Goppa codes. Since a module Groebner basis for a code of this type serves the same purpose as the *generator polynomial* for a cyclic code, and an encoding function can be easily and compactly described in terms of a *division algorithm*. As in [1], (2.3.1) and (2.3.2), we have:

(2.1) Proposition. *Given the Groebner basis $\mathcal{G}$, the information positions and parity check positions for $C_L(D, G)$ are determined as follows:*

a) *The information positions are the coefficients of the non-standard monomials $t^\ell \mathbf{e}_i$, where $\ell \leq |O_i| - 1$ (i.e. monomials divisble by the leading monomial of one of the Groebner basis elements) appearing in the $(h_1(t), \ldots, h_r(t))$ in (2).*

b) *The parity check positions are the standard monomials (i.e. monomials that are not divisible by any of the leading monomials of the Groebner basis elements).*

c) *Fix some enumeration of the information positions – the non-standard monomials m_ℓ. The following algorithm describes a systematic encoder for $C_L(D, G)$:*

```
Input: the Groebner basis $\mathcal{G}$, $w \in \mathbf{F}_q^k$, the list of non-standard monomials $\{m_i\}$
Output: the encoded $E(w) \in C_L(D, G)$
Uses: Normal form (division) algorithm with respect to given ordering
    $f := \sum_{i=1}^k w_i m_i$
    $\overline{f} := \overline{f}^{\mathcal{G}}$ (the normal form)
    $E(w) := f - \overline{f}$
```

In the algorithm, since f is a linear combination of only non-standard monomials, and $\overline{f}$ is a linear combination of only standard monomials, the symbols from w are not changed in the process of computing $E(w)$. Since the normal form of $f - \overline{f}$ is *zero*, $f - \overline{f}$ is an element of the submodule $\overline{C}$, so it represents a codeword.

Here is a Hermitian code example. The technique also applies to a much broader class of codes.

(2.2) Example. For instance, using the *POT* Groebner basis for $C_L(D, 20Q)$ from the Hermitian curve over $\mathbf{F}_9$ from (5) above, we see that

a) the information positions are the coefficents of

$$t^7\mathbf{e}_1, \ldots, t\mathbf{e}_1, \mathbf{e}_1, t^7\mathbf{e}_2, \ldots, t\mathbf{e}_2, t^7\mathbf{e}_3, t^6\mathbf{e}_3, t^5\mathbf{e}_3$$

(18 in all, as expected, since $k = 18$ for this code), and

b) the parity check positions are the coefficients of the remaining monomials:

$$\mathbf{e}_2, t^4\mathbf{e}_3, \ldots, t\mathbf{e}_3, \mathbf{e}_3, t\mathbf{e}_4, \mathbf{e}_4, \mathbf{e}_5$$

By the identification of the codewords with elements of $\mathbf{F}_9[t]^5$, via the orbit structure of the $\mathbf{F}_9$-rational points on the Hermitian curve from (1.2), we can also identify the parity check positions with *a collection of points on the curve.* For $C_L(D, aQ)$ (with the *POT* order as above), the parity check positions are identified with the points:

$$\begin{aligned} P_{2,0} &= (1, \alpha^5), \\ P_{3,4} &= (\alpha^4, \alpha^4), \ldots, P_{3,0} = (1, \alpha^4), \\ P_{4,1} &= (0, \alpha^6), P_{4,0} = (0, \alpha^2), \\ P_{5,0} &= (0, 0) \end{aligned} \tag{9}$$

Using the *POT* Groebner basis as in (5), the encoding operation itself can be organized using several 1-variable polynomial divisions, with respect to the diagonal entries of the generators.

Similarly, the POT information and parity check positions for the $C_L(D, 57\widetilde{Q})$ code on the Hansen-Stichtenoth curve over $\mathbf{F}_8$ can be read off from the form of the Groebner basis (6).

§3. An Application to Decoding.

Up to now, it is certainly fair to say that the major obstacle to the use of AG Goppa codes in practice has been the unavailability of sufficiently efficient decoding algorithms. The *first* generally applicable methods for AG codes from a curve X only corrected

$$t \leq \left\lfloor \frac{d - 1 - g(X)}{2} \right\rfloor$$

errors ([4]). Subsequently, more sophisticated methods were developed that removed the $g(X)$ "penalty," at the cost of introducing more advanced, and less immediately computable, algebraic geometric concepts into the algorithms. But even worse, the "obvious" implementations of the original methods had significantly greater asymptotic complexity (as the length $n \to \infty$) than standard decoding methods for Reed-Solomon, BCH, and "classical" Goppa codes (Berlekamp-Massey, or the generalized Euclidean algorithm). At the present time, the best known decoding algorithm for Hermitian curves is that of [5]. It has an asymptotic complexity of $O(n^{7/3})$. Without better decoding algorithms, the AG codes have not been seen as attractive alternatives.

Most recent work has centered around proposed generalizations of the Berlekamp-Massey method for multi-dimensional cyclic codes, and AG Goppa codes that can be obtained from them. Those methods would apply to a completely general AG Goppa code. For codes with many automorphisms (such as the Hermitian and Hansen-Stichtenoth codes), the module structures identified above may furnish some useful additional information to exploit in seeking such generalizations, although we have nothing concrete to report at present.

It is also natural to ask whether *specialized methods* exploiting the extra symmetry (or equivalently the module structures over various group rings $\mathbf{F}_q[H]$) might also be available. For the remainder of this note, we will present an extended example to show that, at least in some cases, the answer to this question is yes.

The basis for our discussion will be the following elementary fact about syndrome decoding.

(3.1) Lemma. *Let C be an $[n,k,d]$ linear code, with a specific systematic generator matrix $G = [I_k|A]$ (A a $k \times (n-k)$ matrix) and check matrix $H = [-A^t|I_{n-k}]$ (hence a fixed subdivision of the codewords into information positions and parity checks). Let $y = c + e$ be a received word, including an error vector e with $wt(e) \leq t$ ($2t + 1 \leq d$), and let $S = Hy^t$ be the syndrome of y. Then: If $wt(S) \leq t$, then all the information symbols in y are correct, and $e = (0|S)$. On the other hand, if $wt(S) > t$, then at least one information symbol in y is incorrect.*

(See, for instance, [6], p. 512-513.) *Permutation decoding* is possible for C if there is available some set H of automorphisms of the code with the following special property:

(P) For every possible error pattern of weight $t \leq \lfloor \frac{d-1}{2} \rfloor$ there exists some $\tau \in H$ which moves the errors out of the information positions.

If such an H is available, simply apply the automorphisms from H one by one, permuting the entries of the received word, and computing the syndrome of the permuted word, until a syndrome of weight $\lfloor \frac{d-1}{2} \rfloor$ or less is found. Then the errors are all in the parity checks of that permuted word by part a) of the lemma. To decode, one could *re-encode* the correct information symbols from the permuted word, then *"un-permute"* it (apply the inverse automorphism).

First, we will show that permutation decoding is possible for the Hermitian code $C_L(D, 20Q)$ ($[n,k,d] = [27, 18, 7]$), using the *POT* Groebner basis from (5), the resulting identification of information positions and parity checks as points on X_3 from (9), and as H the *entire group* of 216 automorphisms from (7).

That the same result holds for the codes $C_L(D, aQ)$, with $a > 20$ on the Hermitian curve X_3 over $\mathbf{F}_9$ is also clear from our methods, since $d < 7$ for those codes. Moreover for the 1-error-correcting codes in this class it is simple to find a very small set of automorphisms (much smaller than H itself) with the property (P). A similar observation was made already by Tiersma for some 1-error-correcting codes from the Hermitian curve over $\mathbf{F}_4$, in [7].

Given a set T of $\mathbf{F}_9$ rational points on X_3, and a $\tau = \tau_{\lambda,\delta,\mu}$ in H, we will write $\tau(T)$ for the set obtained by applying τ to each element in T.

(3.2) Theorem. *For each set $T \subset X_3(\mathbf{F}_9)$ with $|T| \leq 3$ (the error locations of a correctable error pattern for the $C_L(D, 20Q)$ code), there is some $\tau \in H$ with the property that $\tau(T)$ consists only of points that are POT parity check positions for the code as in (9).*

Idea of Proof: For error patterns of weight 1, this is clear because H acts transitively on the affine $\mathbf{F}_9$-rational points of X_3. For error patterns of weight 2, and error patterns of weight 3 where the three error locations are *collinear* on X_3, the desired conclusion holds because the group H (consisting, as it does, of linear mappings on $\mathbf{F}_9^2$ which restrict to give automorphisms of the Hermitian curve) acts on the set of $\mathbf{F}_9$-rational lines of the affine plane, with *three* orbits: the *vertical lines* $x = c$, the 27 *tangent lines* to X_3 at the $\mathbf{F}_9$-rational points, and the remaining 54 *non-tangent* lines, each of which meets the Hermitian curve in 4 distinct points. Within H, the subgroup fixing a non-tangent line permutes the four points on the line cyclically. So, any T lying entirely on one vertical line can be moved by a $\tau \in H$ to the line $x = 0$ (entirely parity checks). Any T lying entirely on another (necessarily non-tangent) line can be moved to (a subset of) the collinear triple

$$T' = \{P_{3,0} = (1, \alpha^4), P_{3,2} = (\alpha^2, \alpha^4), P_{3,4} = (\alpha^4, \alpha^4)\}$$

in the parity check positions (9).

Finally, we must consider the action of H on *non-collinear* triples of $\mathbf{F}_9$-rational points on the Hermitian curve ("triangles"). Here things get interesting! To understand the structure of this set it is helpful to observe that there is an *invariant* of triangles under the action of H. If $T = \{(x_1, y_1), (x_2, y_2), (x_3, y_3)\}$ is a triangle, then the polynomial function

$$\Delta = (x_1 - x_2)(x_1 - x_3)(x_2 - x_3) \det \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{pmatrix}$$

is invariant under H, and under permutations of the elements of T. ($\Delta = 0$ if and only if there are two points on the same vertical line, and so forth.) In the collection of 2700 triangles, there are:

1) *three* orbits of triangles with $\Delta = 0$, each of size 216,
2) *three* orbits of triangles with $\Delta = 1$, of sizes 36, 72, and 216,
3) *two* orbits each of triangles with $\Delta = \alpha, \alpha^3$, of sizes 216, and 108, and
4) *one* orbit each of triangles with $\Delta = \alpha^2, \alpha^4, \alpha^5, \alpha^6, \alpha^7$, of size 216.

One can check without difficulty that the set of triangles containing only parity check points from (9) contains a triangle from every one of the orbits. For instance the triangle

$$T = \{P_{5,0} = (0, 0), P_{3,4} = (\alpha^4, \alpha^4), P_{3,0} = (1, \alpha^4)\}$$

is in the size-36 orbit of triangles with $\Delta = 1$. (It is analogous to an equilateral triangle in Euclidean geometry, with an isotropy subgroup in H isomorphic to the dihedral group D_3.) $\triangle$

Since each element $\tau = \tau_{\lambda,\delta,\mu}$ of H induces an automorphism of the $C_L(D, aQ)$ codes, the Theorem shows that we have the correct set-up – property (P) above – for permutation decoding, using the entire group H.

A similar result is true for the 3-error-correcting code $C_L(D, 57\widetilde{Q})$ from the Hansen-Stichtenoth curve over $\mathbf{F}_8$.

(3.3) Theorem. *For each set $T \subset Y_1(\mathbf{F}_8)$ with $|T| \leq 3$ (the error locations of a correctable error pattern for the $C_L(D, 57\widetilde{Q})$ code), there is some $\tau \in H$ from (1.6) with the property that $\tau(T)$ consists only of points that are POT parity check positions for the code.*

The details are similar to those for Theorem (3.2), so we will omit them here. To implement this decoding method, the *POT* Groebner bases $\mathcal{G}$ could be used in a decisive way again. Namely, the syndrome computation needed for each permutation of the reserved word could be implemented as a normal form, or division algorithm, computation with respect to $\mathcal{G}$ (the same computation done above in the systematic encoding algorithm (2.1)). The syndrome is nothing other than the remainder on division by $\mathcal{G}$.

Since the groups H here are relatively large, the number of remainder computations necessary to decode a received word by this method makes it less attractive than other known methods for these codes. Nevertheless, it is possible (even likely, I believe) that some smaller subset of H would also suffice for permutation decoding. Determining a *minimal* set of automorphisms with the property (P) needed for permutation decoding (for the a for which it is possible) seems to be, at best, a difficult combinatorial problem.

Unfortunately, it does not seem to be the case that *all* Hermitian 3-error-correcting codes admit permutation decoding, as one might naively hope. Indeed, we have been unable to find *any* monomial ordering for which the parity check positions possess the required property for the Hermitian codes $C_L(D, 56Q)$, $C_L(D, 57Q)$, or $C_L(D, 58Q)$ over $\mathbf{F}_{16}$ ($n = 64, k = 51, 52, 53, d = 8$). So is likely that the examples studied here are quite special in this regard.

References

[1] C. Heegard, J. Little, and K. Saints, "Systematic encoding via Gröbner bases for a class of algebraic geometric Goppa codes." to appear, IEEE Trans. Inform. Theory, 1995.

[2] J. Hansen and H. Stichtenoth, "Group codes on certain algebraic curves with many rational points," *Applicable Algebra in Eng. Comm. Comp.*, vol. 1, pp. 67–77, 1990.

[3] H. Stichtenoth, *Algebraic Function Fields and Codes.* Berlin: Springer, 1993.

[4] A. N. Skorobogatov and S. G. Vlăduţ, "On the decoding of algebraic-geometric codes," *IEEE Trans. Inform. Theory*, vol. 36, pp. 1051–1060, Sept. 1990.

[5] J. Justesen, K. J. Larsen, H. Elbrønd Jensen, and T. Høholdt, "Fast decoding of codes from algebraic plane curves," *IEEE Trans. Inform. Theory*, vol. 38, pp. 111–119, Jan. 1992.

[6] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes.* Amsterdam: North-Holland, 1977.

[7] H. Tiersma, "Remarks on codes from Hermitian curves," *IEEE Trans. Inform. Theory*, vol. 33, pp. 605–609, 1987.

Supervisory Control of Finite Automata Under Dynamic Predicate Specifications

S. L. Chen W. M. Wonham
Systems Control Group
Dept. of Electrical & Computer Eng.
Univ. of Toronto
email: chensl@control.utoronto.ca wonham@control.utoronto.ca

Abstract: This paper discusses supervisory control of finite automata under dynamic predicate specifications. Controllers are represented as Vector Discrete Event Systems (VDES). The controlled plants are Discrete Event Systems with mixed structural type (MDES).

I. Introduction

Several Discrete Event System (DES) model types have been suggested and investigated over the last decade ([1]). However, the existing models are all of homogeneous structure:, that is their state space is defined over either an abstract node set or a nonnegative integer vector set. In the real world, physical plants may consist of subplants with different structural properties, so DES models of homogeneous structural type are not always adequate.

In this paper, we combine different structures into a unified DES model. Open-loop plants are given as finite automata (RW model) ([4]), control specifications as dynamic predicates, and controllers enforcing the specifications as Vector Discrete Event Systems (VDES) ([2] [3]). As such the controlled plants are DES with mixed structural type (MDES).

II. Finite Automata Models

A finite automaton model is defined as ([4]) $G_R := (\mathbf{Q},\ \Sigma_R,\ \xi_Q,\ q_0,\ \mathbf{Q_m})$. Here Σ_R is a finite event set partitioned into two subsets, namely, the uncontrollable event set denoted by $\Sigma_{R,u}$, and the controllable event set denoted by $\Sigma_{R,c}$ of which each event can be disabled by a controller. $\mathbf{Q} = \{q_0,\ q_1,\ \cdots,\ q_m\}$ is the state set, q_0 is the initial state, $\mathbf{Q_m} \subseteq \mathbf{Q}$ is the subset of marker states. In this paper we assume $\mathbf{Q_m} = \mathbf{Q}$.

$\xi_Q : \mathbf{Q} \times \Sigma_R \to \mathbf{Q}$ is the (partial) transition function. For $q \in \mathbf{Q}$ and $\sigma \in \Sigma_R$, $\xi_Q(q,\sigma)!$ means $\xi_Q(q,\sigma)$ is defined. If $\xi_Q(q,\sigma)!$, we say σ is enabled by the plant at state q. The function can be extended to $\mathbf{Q} \times \Sigma_R^*$ by

$$\xi_Q(q,\epsilon) := q,\ \ \xi_Q(q,s\sigma) := \xi_Q(\xi_Q(q,s),\sigma), \qquad \text{for } s \in \Sigma_R^*,\ \sigma \in \Sigma_R.$$

Here ϵ is the empty string. The behaviour of G_R is described by its language , defined as $L(G_R) = \{\ s \mid s \in \Sigma_R^*,\ \xi_Q(q_0,s)!\}$

Given a finite automaton G_R; for $q \in \mathbf{Q}$ and $\sigma \in \Sigma_R$, we define :

$$q_R^{\downarrow} = \{\sigma \mid \xi_Q(q,\sigma)!\}, \qquad q_R^{\uparrow} = \{\sigma \mid (\exists q')\ \ \xi_Q(q',\sigma) = q\}, \qquad \sigma_R^{\uparrow} = \{q \mid \xi_Q(q,\sigma)!\}.$$

III. Vector Discrete Event Systems

A Vector Discrete Event System is defined as ([2][3]) $G_v := (\mathbf{X},\ \Sigma_v,\ \xi_X,\ x_0)$ Here $\mathbf{X} = \{x = [x_1\ x_2\ \cdots\ x_n] \mid x_i \in \mathbb{N} \text{ for } i = 1 \cdots n\} = \mathbb{N}^n$ is the state space of G_v; x_0 is the initial state. Each state x is described by the state variables $x_1,\ x_2,\ \cdots,\ x_n$. Σ_v is a finite event set partitioned into the uncontrollable event subset denoted by $\Sigma_{v,u}$ and controllable event subset denoted by $\Sigma_{v,c}$. Each $\sigma \in \Sigma_v$ is associated with an n-dimensional integer vector e_σ, the *displacement vector* of σ. The vector describes the action of σ on a state x.

$\xi_X : \mathbf{X} \times \Sigma_v \to \mathbf{X}$ is the (partial) transition function. For $x \in \mathbf{X}$ and $\sigma \in \Sigma_v$, $\xi_X(x,\sigma)$ is defined, denoted by $\xi_X(x,\sigma)!$, if and only if $x + e_\sigma \geq 0$. If $\xi_X(x,\sigma)!$ then $\xi_X(x,\sigma) := x + e_\sigma$, and we say σ is enabled by the plant at state x. The function can be extended to $\mathbf{X} \times \Sigma_v^*$ as ξ_Q. The behaviour of G_v is described by its language , defined as

$$L(G_v) = \{\ s \mid s \in \Sigma_v^*,\ \xi_X(x_0, s)!\}.$$

IV. Discrete Event Systems of Mixed Structural Type (MDES)

An MDES is defined as $G_M := G_R||_s G_v = (\mathbf{Q} \times \mathbf{X},\ \Sigma,\ \xi,\ (q_0, x_0))$. Here $\mathbf{Q}$ is the symbolic state set of which each element is called a symbolic state; q_0 is the initial symbolic state. $\mathbf{X}$ is the numerical state space of which each element is called a numerical state; x_0 is the nonnegative initial numerical state.

$\Sigma := \Sigma_R \cup \Sigma_v$ is a finite event set. Events in $\Sigma_R \cap \Sigma_v$ are called shared events. The event set is partitioned into the uncontrollable event set denoted by Σ_u and the controllable event set denoted by Σ_c. We define $\Sigma_u := \Sigma_{R,u} \cup \Sigma_{v,u}$, and $\Sigma_c := \Sigma - \Sigma_u$. That is, an event σ is uncontrollable in G_M if it is uncontrollable in either G_R or G_v.

$\xi : \mathbf{Q} \times \mathbf{X} \times \Sigma \to \mathbf{Q} \times \mathbf{X}$ is the partial transition function. For $(q, x) \in \mathbf{Q} \times \mathbf{X}$ and $\sigma \in \Sigma$,

$$\xi(q, x, \sigma) := \begin{cases} (\xi_Q(q,\sigma), x) & \sigma \in \Sigma_R - \Sigma_v \\ (q, \xi_X(x,\sigma)) & \sigma \in \Sigma_v - \Sigma_R \\ (\xi_Q(q,\sigma), \xi_X(x,\sigma)) & \sigma \in \Sigma_R \cap \Sigma_v \end{cases}$$

By definition of the partial transition function, a shared event can occur if only if it is enabled by both G_R and G_v. The behaviour of a MDES G_M is described by its language

$$L(G_M) = \{s \mid s \in \Sigma^*\ \ \xi(q_0, x_0, s)!\}.$$

V. Dynamic Predicates

Given a plant G_R with event set $\Sigma_R = \{\sigma_1,\ \sigma_2,\ \cdots,\ \sigma_n\}$. For a string $s \in L(G_R)$, define its occurrence vector $v_s : n \times 1$ as $v_s(j) = |s|_{\sigma_j}$ where $|s|_{\sigma_j}$ is the number of occurrences of σ_j in s. Define $\mathbf{V} := \{v_s \mid s \in L(G_R)\ \} \subseteq \mathbb{N}^n$, the occurrence space of the plant G_R. A dynamic predicate P_{dyn} on $\mathbf{V}$ is defined as a condition on $v \in \mathbf{V}$ of the form:

$$cv \leq d \qquad c : 1 \times n \qquad v : n \times 1 \qquad d : \text{nonnegative integer}$$

where c is a constant integer vector and $v(i) = |s|_{\sigma_i}$ for $s \in L(G_R)$. Sometimes a P_{dyn} is written as $\sum_{i=1}^{n} c_i|\sigma_i| \leq d$, where c_i is the component of c corresponding to σ_i and

$|\sigma_i| := v(i)$. Also we may write c_{σ_i} instead of c_i.

Given a control specification P_{dyn}; define

$$\mathrm{L}(\mathrm{G}_\mathrm{R},\ P_{dyn}) = \{\ s \in \mathrm{L}(\mathrm{G}_\mathrm{R})\ \mid\ (\forall t \leq s)\ cv_t \leq d\}.$$

The set contains all the feasible strings whose prefixes satisfy the predicate. Our control objective is to find a supervisory control ([4]) s.t. the controlled behaviour of the plant satisfies P_{dyn}. Also the control is *optimal* i.e. the language generated by the controlled plant is the largest controllable sublanguage of $\mathrm{L}(\mathrm{G}_\mathrm{R},\ P_{dyn})$, denoted by $\sup C(P_{dyn})$. We can easily prove that

$$\sup C(P_{dyn}) = \{\ s \in \mathrm{L}(\mathrm{G}_\mathrm{R})\ \mid\ (\forall s' \leq s)(\forall l \in \Sigma^*_{\mathrm{R},u})\ s'l \in \mathrm{L}(\mathrm{G}_\mathrm{R}) \Rightarrow cv_{s'l} \leq d\}.$$

VI. Weighted Automata

Given a plant G_R and a dynamic predicate specification $P_{dyn}:\ cv \leq d$. The plant can be treated as a weighted automaton s.t. each event σ is associated with a weight $w(\sigma) := c_\sigma$. A string, $s := \sigma_1\sigma_2\cdots\sigma_n$, has weight $w(s) := \sum_{i=1}^{n} w(\sigma_i)$. With a slight abuse of terminology, we call a string s a *path* from node q to q$'$ if $\xi_\mathrm{Q}(\mathrm{q}, s) = \mathrm{q}'$. Also, if $\xi_\mathrm{Q}(\mathrm{q}, \sigma) = \mathrm{q}'$, we call σ an *edge* from q to q$'$.

Let $\mathbf{Q} = \{\mathrm{q}_0,\ \mathrm{q}_1,\ \cdots,\ \mathrm{q}_{n-1}\}$. We define an $n \times n$ matrix M, called the *associated matrix of* G_R, s.t. $(\forall i, j \in \{0,\ 1,\ \cdots,\ n-1\})$

$$\mathrm{M}(ij) = max\ \{w(\sigma)\ \mid\ \xi_\mathrm{Q}(\mathrm{q}_\mathrm{i}, \sigma) = \mathrm{q}_\mathrm{j}\ \}$$

with $\mathrm{M}(ij) = -\infty$ if the defining set is empty.

Denote by $\overline{\mathbf{Z}}$ the extended integer set $\mathbf{Z} \cup \{+\infty,\ -\infty\}$ and define two operators

$$\oplus : \overline{\mathbf{Z}} \times \overline{\mathbf{Z}} \longrightarrow \overline{\mathbf{Z}}$$

$$\otimes : \overline{\mathbf{Z}} \times \overline{\mathbf{Z}} \longrightarrow \overline{\mathbf{Z}}$$

according to:
(1) $(\forall a,\ b)\ \ a \oplus b = max(a,\ b)$.
(2) $(\forall a,\ b)\ \ a \otimes b = a + b$.

Next define

$$\times : \overline{\mathbf{Z}}^{n \times n} \times \overline{\mathbf{Z}}^{n \times n} \longrightarrow \overline{\mathbf{Z}}^{n \times n}$$

by $(\forall \mathrm{M},\ \mathrm{N} \in \overline{\mathbf{Z}}^{n \times n})\ \ \ (\forall i, j \in \{0,\ 1,\ \cdots,\ n-1\})$:

$$(\mathrm{M} \times \mathrm{N})(ij) = (\mathrm{M}(i0) \otimes \mathrm{N}(0j)) \oplus (\mathrm{M}(i1) \otimes \mathrm{N}(1j)) \oplus \cdots \oplus (\mathrm{M}(i(n-1)) \otimes \mathrm{N}((n-1)j))$$

Define $(\forall \mathrm{M} \in \overline{\mathbf{Z}}^{n \times n})\ (\forall k \geq 2)\quad \mathrm{M}^k = \mathrm{M} \times \mathrm{M}^{k-1}$.

If $\mathrm{M}^m(ij) = -\infty$ $(m \in \mathbf{Z}_+)$ then there is no m-edge path from node q_i to node q_j. If $\mathrm{M}^m(ij) = w \in \mathbf{Z}$ then w is the maximum weight among all the m-edge paths from q_i to q_j.

Let $\mathrm{q} \in \mathrm{Q}$; define $\mathrm{L_q} := \{s \in \Sigma^*_{\mathrm{R},u} \mid \xi_{\mathrm{Q}}(\mathrm{q}, s)!\}$, i.e., $\mathrm{L_q}$ is the set of uncontrollable strings feasible at q. If $\mathrm{L_q}$ is nonempty, we define a function

$$f_{\mathrm{q}} : \mathrm{L_q} \to \mathbf{Z}$$

s.t. $(\forall s \in \mathrm{L_q})\ f_{\mathrm{q}}(s) = w(s)$.

If f_{q} has no finite upper bound then the given specification P_{dyn} can be enforced only if state q is never entered. If f_{q} has a finite upper bound then it has a least upper bound $sup\ f_{\mathrm{q}} := m$ s.t. m is the maximum weight that all the uncontrollable strings feasible at q can achieve. If m is nonpositive then it is safe to enter q. If m is positive then we have to prevent entrance to q by supervisory control, which we shall implement as a VDES controller.

Let $\mathrm{G_{R,sub}}$ be the restriction of $\mathrm{G_R}$ to $\Sigma_{\mathrm{R},u}$, and M the associated matrix of $\mathrm{G_{R,sub}}$; then we have the following results.

Theorem 1: $f_{\mathrm{q_i}}$ has no finite upper bound iff $(\exists\ 1 \leq k_1, k_2 \leq n,\ \ 0 \leq j \leq n-1)$, s.t. $\mathrm{M}^{k_1}(ij) \in \mathbf{Z}$ and $\mathrm{M}^{k_2}(jj) \in \mathbf{Z}_+$(positive integers), where $n = |\mathbf{Q}|$.

Theorem 2: If $f_{\mathrm{q_i}}$ has a finite upper bound then it has a least upper bound $sup\ f_{\mathrm{q_i}} = m$ s.t. $(\exists 1 \leq k \leq n,\ \ 0 \leq j \leq n-1)\ \ \mathrm{M}^k(ij) = m$.

An efficient algorithm has been developed to check whether f_{q} has a finite upper bound and compute the least upper bound if it exists.

VII. Dynamic Predicate Operators

For $\mathrm{q} \in \mathbf{Q}$, if $\mathrm{L_q}$ is nonempty and f_{q} has a least upper bound m, we define a dynamic predicate operator T_{q} : *set of dynamic predicates* $\rightarrow$ *set of dynamic predicates* as:

$$P_{new,dyn} := T_{\mathrm{q}} P_{dyn}$$

where $P_{new,dyn} := (c_{new} v \leq d_{new})$ is:
(1) if $m \leq 0$ then $P_{new,dyn} := P_{dyn}$ i.e. T_{q} is the identity operator;
(2) if $m > 0$ then
(2.1) if q is the initial state of $\mathrm{G_R}$, $\mathrm{d_{new}} := d - m$; else $d_{new} := d$;

$$(2.2)\ c_{new,\sigma} := \begin{cases} c_\sigma - m & \text{if } \sigma \in (\mathrm{q}_{\mathrm{R}}^{\downarrow} - \mathrm{q}_{\mathrm{R}}^{\uparrow}) \\ c_\sigma + m & \text{if } \sigma \in (\mathrm{q}_{\mathrm{R}}^{\uparrow} - \mathrm{q}_{\mathrm{R}}^{\downarrow}) \\ c_\sigma & \text{otherwise} \end{cases}$$

If $d_{new} \geq 0$ then accept $P_{new,dyn}$ with d_{new} and $c_{new,\sigma}$; otherwise $P_{new,dyn} :=$ **NULL** s.t. $\sup C(\mathbf{NULL}) := \emptyset$.

A negative d_{new} means that there is an uncontrollable string s feasible at the initial state such that $cv_s > d$. In fact we can easily prove that $\sup C(P_{dyn}) = \emptyset$ if $d_{new} < 0$.

Theorem 3: $\sup C(P_{new,dyn})$=$\sup C(P_{dyn})$.

The last theorem says that the dynamic predicate operator preserves the supremal controllable sublanguage. We will use the operator to compute a VDES controller enforcing P_{dyn}.

Theorem 4: $(\forall\sigma \in \Sigma_{\mathrm{R},u})$, if $\xi_{\mathrm{Q}}(\mathrm{q},\sigma) = \mathrm{q}'$ and $P_{new,dyn} := T_{\mathrm{q}}T_{\mathrm{q}'}P_{dyn}$ (or $P_{new,dyn} := T_{\mathrm{q}'}T_{\mathrm{q}}P_{dyn}$), then $c_{new,\sigma} \leq 0$.

Theorem 4 guarantees that if σ is uncontrollable then c_σ will eventually become nonpositive.

VIII. Existence and Computation of VDES Controllers

This section presents the main result of the paper.

Theorem 5: Given a plant $\mathrm{G_R}$ and a specification P_{dyn}, if $\mathrm{G_R}$ is uniquely labeled (i.e $\forall\sigma$, $\sigma_{\mathrm{R}}^{\uparrow}$ is a singleton) there exists a VDES controller $\mathrm{G_v}$ s.t. $\mathrm{L}(\mathrm{G_R}||_s\mathrm{G_v}) = \sup C(P_{dyn})$. The controlled plant is an MDES.

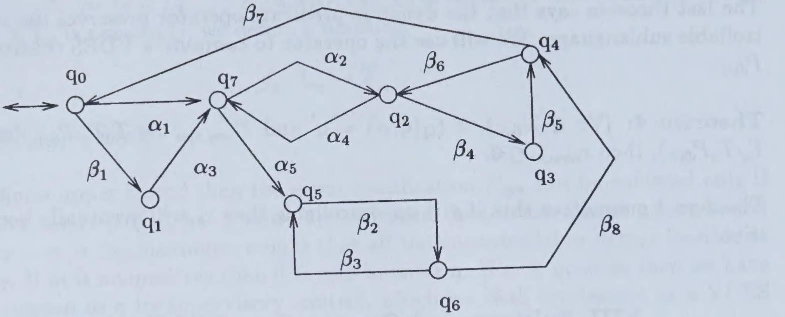
The unique labeling condition imposed on $\mathrm{G_R}$ can be weakened. In fact only a subsystem of $\mathrm{G_R}$ is required to be uniquely labeled, and the subsystem can be computed systematically. Due to limitation of space, we present Theorem 5 only in simplified form.

An efficient procedure is developed to compute the controller $\mathrm{G_v}$. Here is the main idea. We compute *sup* f_{q} for each $\mathrm{q} \in \mathrm{Q}$. If f_{q} is unbounded, we find all the controllable events whose occurrence may enable some uncontrollable events via which the plant will enter the state q. We define a VDES $\mathrm{G_{v,dis}}$ to disable these controllable events by setting a zero initial condition, negative displacement vectors for these controllable events and zero displacement vectors for other events. If f_{q} is bounded by m then we apply T_{q} to P_{dyn}. Since each plant has a finite number of states, we only apply a finite number of operators to P_{dyn}. As such $\sup C(P_{dyn})$ is preserved. Eventually, we obtain a new dynamic predicate $P_{new,dyn}$ s.t. $(\forall\sigma \in \Sigma_{\mathrm{R},u})$, $c_{new,\sigma} \leq 0$ by THeorem 4. According to result in [3], the new predicate is equivalent to a VDES $\mathrm{G_{v,con}}$ in which the initial state is d_{new}, and $e_\sigma := -c_{new,\sigma}$. Also $\mathrm{G_{v,con}}$ is controllable w.r.t. $\mathrm{G_R}$ since all the uncontrollable events have nonnegative displacement vectors. The controller VDES $\mathrm{G_v}$ is the synchronous product [2] of $\mathrm{G_{v,con}}$ and $\mathrm{G_{v,dis}}$.

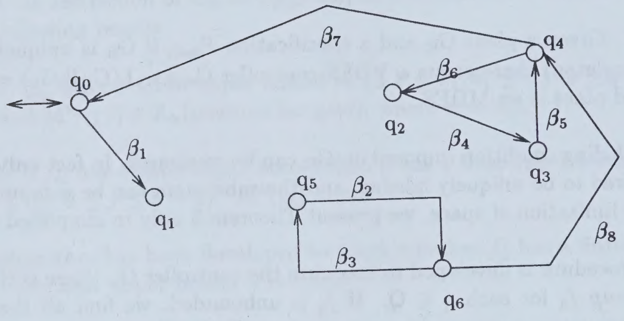
Let's look at the following tutorial example.

Example 1: Consider the plant $\mathrm{G_R}$ given in Fig. 1. Assume that all the α events are controllable and all the β events are uncontrollable. Let the specification be

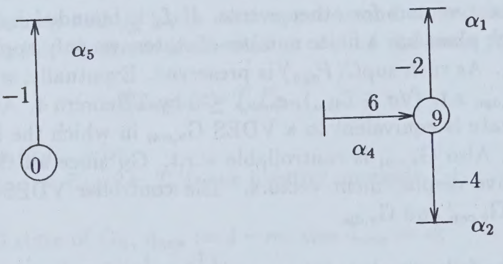
$$P_{dyn} \quad 3|\alpha_1| + 2|\alpha_2| - 4|\alpha_4| + 5|\alpha_5| + |\beta_1| + |\beta_2| + |\beta_4| - |\beta_5| + |\beta_7| - |\beta_8| \leq 1.$$


q0
q7
q2
q4
q3
q1
q5
q6
β7
α2
β6
α1
α4
β5
β4
β1
α3
α5
β2
β3
β8

G_R


q0
q4
q2
q3
q1
q5
q6
β7
β6
β5
β4
β1
β2
β3
β8

$G_{R,sub}$


α5
−1
0
α1
−2
6
9
α4
−4
α2

$G_{v,dis}$ $G_{v,con}$

Fig. 1 Example 1

The subsystem $G_{R,sub}$ is shown in Fig. 1. Its associated matrix is M. Since there are 7 nodes in the subsystem, we need to compute M^2 up to M^7. For brevity, only M and M^2 are displayed. In the matrices, ϵ denotes $-\infty$.

$$\mathrm{M} = \begin{bmatrix} \epsilon & 1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 1 & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & -1 & \epsilon & \epsilon \\ 1 & \epsilon & 0 & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 1 \\ \epsilon & \epsilon & \epsilon & \epsilon & -1 & 0 & \epsilon \end{bmatrix} \qquad \mathrm{M}^2 = \begin{bmatrix} \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & 0 & \epsilon & \epsilon \\ 0 & \epsilon & -1 & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & 2 & \epsilon & 1 & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & 0 & \epsilon & \epsilon \\ 0 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 1 \end{bmatrix}$$

We write f_i for $f_{\mathrm{q_i}}$ in the later discussions.

By Theorem 1, we know that f_5 and f_6 have no finite upper bound. So q_5 and q_6 should never be entered. As a consequence, controllable event α_5 should never be enabled. We define a VDES controller $\mathrm{G_{v,dis}}$, shown in Fig. 1, to disable α_5. Since the initial condition of $\mathrm{G_{v,dis}}$ is zero and α_5 has a negative displacement vector, α_5 will never occur in $\mathrm{G_{v,dis}}$.

By examining all the matrices, we have $sup\ f_0 = 1$, $sup\ f_2 = 2$, $sup\ f_3 = 1$ and $sup\ f_4 = 2$. Let $P_{new,dyn} := T_4T_3T_2T_0P_{dyn}$; then

$c_{new,\alpha_1} = c_{\alpha_1} - sup\ f_0 = 2$

$c_{new,\alpha_2} = c_{\alpha_2} + sup\ f_2 = 4$

$c_{new,\alpha_3} = c_{\alpha_3} = 0$

$c_{new,\alpha_4} = c_{\alpha_4} - sup\ f_2 = -6$

$c_{new,\alpha_5} = c_{\alpha_5} = 0$

$c_{new,\beta_1} = c_{\beta_1} - sup\ f_0 = 0$

$c_{new,\beta_2} = c_{\beta_2} = 1$

$c_{new,\beta_3} = c_{\beta_3} = 0$

$c_{new,\beta_4} = c_{\beta_4} - sup\ f_2 + sup\ f_3 = 0$

$c_{new,\beta_5} = c_{\beta_5} - sup\ f_3 + sup\ f_4 = 0$

$c_{new,\beta_6} = c_{\beta_6} - sup\ f_4 + sup\ f_2 = 0$

$c_{new,\beta_7} = c_{\beta_7} + sup\ f_1 - sup\ f_4 = 0$

$c_{new,\beta_8} = c_{\beta_8} + sup\ f_4 = 0$

$d_{new} = d - sup\ f_0 = 9$

Since state q_5 will never be visited, β_2 is permanently disabled and we set $c_{new,\beta_2} := 0$.

The new predicate $P_{new,dyn}$ is

$$4|\alpha_2| + 2|\alpha_1| - 6|\alpha_4| \leq 9$$

The new predicate $P_{new,dyn}$ is equivalent to a VDES $\mathrm{G_{v,con}}$ as shown in Fig. 1. Since all the uncontrollable events have zero displacement vectors in $\mathrm{G_{v,con}}$, $\mathrm{G_{v,con}}$ is controllable w.r.t. $\mathrm{G_R}$. The VDES controller $\mathrm{G_v}$ is the synchronous product of $\mathrm{G_{v,con}}$ and $\mathrm{G_{v,dis}}$.

Corollary 6: Given two automata $\mathrm{G_{R,1}}$ with an event set $\Sigma_{\mathrm{R,1}}$ and $\mathrm{G_{R,2}}$ with an event set $\Sigma_{\mathrm{R,2}}$. If both $\mathrm{G_{R,1}}$ and $\mathrm{G_{R,2}}$ are uniquely labeled and $\Sigma_{\mathrm{R,1}} \cap \Sigma_{\mathrm{R,2}} = \emptyset$, then there exists a VDES controller $\mathrm{G_v}$ s.t. $\mathrm{L(G_R||_sG_v)} = \sup C(P_{dyn})$ where $\mathrm{G_R}$ is the shuffle product ([4]) of $\mathrm{G_{R,1}}$ and $\mathrm{G_{R,2}}$. The controlled plant is an MDES.

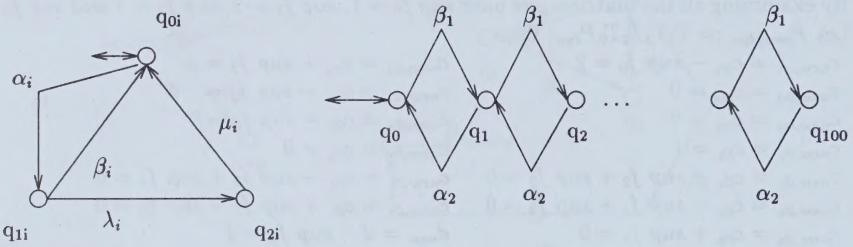
A procedure is developed to compute $\mathrm{G_v}$. Using the procedure, we don't need to compute $\mathrm{G_R}$ and state explosion problem can be avoided.

IX. Control of a Production Cell

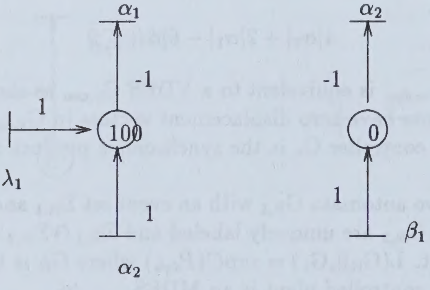
In this section an application example is presented.

Example 2: Consider a small factory with two machines MACH1 and MACH2, and a buffer with 100 slots. Small Factory operates as follows. Initially the buffer is empty,. With the event α_1, MACH1 takes a workpiece from an infinite input bin and enters q_{11} (working state). Subsequently MACH1 either breaks down (event λ_1)and enters q_{21} , or successfully completes its work cycle, deposits the workpiece in the buffer (event β_1), and returns to the initial state q_{01}. MACH2 operates similarly, but takes its workpiece from the buffer and deposits it when finished in an infinite output bin. If a machine breaks down, then on repair it returns to its initial state (event $\mu_i \ \ i = 1,\ 2$).

The control specification is that the buffer must not overflow or underflow. In the RW framework, plants and the specification are modeled by finite automata as shown in Fig. 2. Obviously, both MACH1 and MACH2 satisfy Corollary 6. Assume β_1, β_2, λ_1 and λ_2 are uncontrollable. The RW model for the specification is augmented by a list of self-loop events α_1, λ_1, μ_1, β_2 and μ_2 . The RW controller can be computed by using TCT software package. The controller has hundreds of states and transitions.



MACH$_i$ RW SPEC.



VDES Controller 1 VDES Controller 2

Fig. 2 Example 2

The specification can be represented as two dynamic predicates:

$$P_{dyn1}: \quad |\beta_1| - |\alpha_2| \leq 100 \qquad \text{no overflow}$$

$$P_{dyn2}: \ |\alpha_2| - |\beta_1| \leq 0 \quad \text{no underflow}$$

Two VDES controllers enforcing the specification are shown in Fig. 2. Obviously, they have a simpler structure than the RW controller.

Suppose we have a larger buffer now, say, with 200 slots, and the specification is modified accordingly. In the RW framework, we have to set up a new model for the modified specification, and re-compute the controller. However, in the MDES framework, the only thing we need to do is to change the initial state of Controller 1 to 200.

The usefulness of MDES is clearly presented in the example.

References

[1] Cassandras C.G., 1993 "Discrete Event Systems: modeling and performance analysis", Aksen Associates, Inc. MA.

[2] N.Q. Huang, 1991 "Supervisory Control of Vector Discrete-Event Systems", M.A.Sc. Thesis, Dept. of Electrical Eng., University of Toronto.

[3] Y. Li, 1991 "Control of Vector Discrete-Event Systems", Ph.D. Thesis, Dept. of Electrical Eng., University of Toronto.

[4] P.J. Ramadge and W.M. Wonham, 1989 "The Control of Discrete-Event Systems", Proceeding IEEE 77(1), pp. 81-98.

The Design of Corridor Policies with Separable Average Criteria *

Carlos Humes Jr. - Leônidas de Oliveira Brandão - Manuel Pera Garcia

Instituto de Matemática e Estatística - Universidade de São Paulo

Caixa Postal 66281 (ag. Iguatemi), CEP 05389-970, Brazil[†]

Abstract

In this work we address the question of finding optimal corridor policies for systems evaluated by a separable average criterion. Besides solving such problem, this work shows that there are several solutions to the problem, solutions that independ of the trajectory to be tracked and that we can have several solutions with negligible transients.

This indicates that linear corridor policies can be viewed as a class of scheduling policies with limit cycle, with good transient behaviour and easy to be designed with respect to the separable average criteria.

1 Introduction

The problem of finding optimal scheduling policies for manufacturing systems has several sources of mathematical complexities. To name just two of them, we note that set-up times introduce delays in the description of the dynamical behaviour of the system and that the indivisibility of the workstations lead to integer variables, source of complexity in optimization problems. This verification leads naturally to the idea of using decomposition techniques of the type relaxation/restriction in order to handle the so-called "complicating variables" ([4]).

The idea of hierarchical control ([1]) can be seen as the usage of two steps in such decomposition framework. The first step (*high level planning*) corresponds to the solution of a "relaxed problem", that does not consider, for instance, set-up times and indivisibility. The next step (*tracking*) is a restriction step, where we try to replicate the relaxed optimal trajectory in a model that considers the "complicating variables". If the tracking solution allow us to follow precisely the relaxed solution, then the solution of the restricted problem is the solution for the optimal scheduling problem. Although this is not generally the case, the solution obtained in the tracking step is usually considered as being a good feasible solution.

Corridor policies are typical of this framework and are used for the scheduling of one machine subject to demands described by linear accumulated demand functions. The machines do not present self loops (i.e. each task in the machine feeds directly a

---

*This research has been partially funded by FAPESP (Proc.No.93/0603-1 and 94/1376-1), CNPq, IBM-Brasil and NFS (grant ECS-9403571).

[†] Email address: `humes@ime.usp.br`, `leo@ime.usp.br` and `mane@ime.usp.br`

demand or, equivalently, each part is processed exactly once in the machine), and the optimal relaxed trajectory has a target point, this is, for the relaxed problem there is an optimal operating point. This last assumption is quite reasonable if our objective function corresponds to the minimization of costs associated to inventory levels and unfullfiled demands ([2] and [3]).

In the literature, the study of corridor policies concentrates upon linear corridor policies characterized by a polyhedrical cone with vertex at the target point and by an operating rule that determines that a production change occurs when a facet of the cone is reached.

It can be shown that a convenient choice of the cone will induced the existence of a *limit cycle* (if the system is stable [3] and [8]), otherwise, even if the system is stable its dynamics will be chaotic ([9] and [10]).

The purpose of this paper is to present a methodology for the design of optimal corridor policies. Among others, two surprising results appear. First of all, if we constrain ourselves to operating the machine with corridor policies, the optimal corridor policy may be independent of the trajectory to be tracked. Moreover, in general, the optimal corridor policy is associated to a cone whose vertex is not the target point of the trajectory obtained in the solution of the relaxed problem.

The paper is organized as follows. In section 2, we present the model in consideration following the lines of the pioneer work of Sharifnia, Caramanis & Gershwin. In section 3, we present alternative descriptions of the systems dynamics and review stability results. In section 4, we present the core of the results for design of corridor policies. In section 5, we present comments and open questions related to this work.

2 The setting

Consider a machine (or plant) that processes n products, each of them with a linear accumulated demand described as

$$D_j(t) = d_j t, \qquad j = 1, 2, \ldots, n.$$

We shall refer do d_j as input rate of product j.

Suppose that each product is serviced precisely once by the machine and the time required to process one unit of product j is constant and denoted by τ_j $(j = 1, 2, \ldots, n)$.

When the machine is processing product j, we say it is devoted to task j and the machine can not execute more than one task at a time (indivisibility). Moreover, we consider that the process of changing tasks to be serviced (from task i to task j) leads to a set-up time ($\delta_{ij} \geqq 0$) correspondent to a period of time when no actual production occurs.

In this setting, we may choose as state variables the accumulated excess production for all tasks ($x_j(t), j = 1, 2, \ldots, n$). The dynamics of the system can be described as

$$\dot{x}(t) = u \in \{v^0, v^1, \ldots, v^n\}$$

$$\text{where} \left\{ \begin{array}{lcl} u & = & v^0 = -d, \text{ if at time } t \text{ the system is idle or in a set-up period;} \\ u & = & v^i = (1/\tau_j)e^j - d, \text{if at time } t \text{ the system is actually producing product } j. \end{array} \right.$$

So, using $\tau_M = [\tau_1 e^1 | \tau_2 e^2 | \cdots | \tau_n e^n]$, and defining $u_j(t) = 1$, if at time t the system is actually servicing task j, otherwise, $u_j(t) = 0$, we obtain

$$\dot{x}(t) = \tau_M^{-1}u(t) - d.$$

Remembering that d_j is the input rate of product j and that τ_j is the processing time of one unit of such product, we define the congestion level of task j as $\rho_j = d_j\tau_j$. This information can be aggregated in the vector $\vec{\rho} = (\rho_1, \rho_2, \cdots, \rho_n)$. The congestion level of the machine is defined as

$$\rho =< 1, \vec{\rho} >= \sum_{j=1}^{n} \rho_j.$$

A basic assumption used throughout this work is the *capacity condition:* $\rho < 1$.

For this model, convex corridor policies are defined by

(i) a cone $C(A,b) = \{x \in \mathbb{R}^n | Ax \leqq b\} = \{x \in \mathbb{R}^n | < a_i, x > \leqq b_i,\ i = 1, 2, \ldots, n\}$;

(ii) the operating rule: "once the trajectory reaches a facet of the cone $C(A,b)$, defined by $< a_k, x >= b_k$, the scheduler starts a set-up towards the production of product $(k \ mod \ n) + 1$".

Given the above operating rule, the trajectory is determined by the knowledge of the values of $m_{ij} = < a^i, v^j >$ $(i,j = 1, \ldots, n)$, i.e., by the matrix $M = AV$, where $V = [v^1 \mid \cdots \mid v^n]$.

It has been show ([3], [8]) that a sufficient condition for the sequence of controls to be periodic is that $\{m_{ij} > 0 \Leftrightarrow i = j\}$ and $\left\{Ad = (1-\rho)^{-1}M\vec{\rho} \geqq \vec{0}\right\}$. This condition, denoted $M \in \mathcal{S}$, will be assumed throughout this work. The necessity of this condition is discussed in some recent works ([9], [10]). It must be noted that the periodicity of the sequence of the controls does not imply in a periodic trajectory.

We must impose $\left\{M\vec{\rho} \geqq \vec{0}\right\}$ in order to guarantee that the trajectory does not leave the cone and then the operating rule (ii) is well defined and *we shall do so.* If we did not wish to impose such constraint, in (ii) we should have substituted "reaches a facet" by "reaches a facet from within the cone", but a number of technicalities should be faced ([10]).

3 The dynamics induced by $C(A,b)$

First of all, as we constrain ourselves to corridor policies where the sequence of controls is periodic, there is no loss of generality in assuming that the products will be serviced in a round-robin sequence $\{1, 2, \ldots, n\}$ $(M \in \mathcal{S})$. For this sequence, the set-ups can be indicated by δ_j the set-up times for the production of product j.

Denoting the duration of the kth actual servicing of task j by Δ_j^k and by x^k the state of the system when the kth set-up time for task 1 is initiated (t^k), in [3] it has been shown that the trajectory can be represented in the form

$$x^{k+1} = Bx^k + Cs + Db,$$

where B, C and D have a non trivial description.

A somewhat simpler description can be obtained if we study the system dynamics for state vector described in the basis $\{v^1, v^2, \ldots, v^n\}$, i.e., $y = V^{-1}x$ ([6]),

$$y^{k+1} = Jy^k + L^{-1}b + L^{-1}SM\vec{\rho} - (1-\rho)^{-1}\|\delta\|_1\vec{\rho}, \tag{1}$$

$$\text{where} \begin{cases} M &= L+U,\ l_{ij}=0, \text{ for } i<j, \text{ and } u_{ij}=0, \text{ for } i \geqq j; \\ J &= L^{-1}U; \\ S &= (1-\rho)^{-1} diag(\delta_1, \delta_1+\delta_2, \ldots, \sum_{j=1}^{n}\delta_j). \end{cases}$$

Another quite similar representation is obtained using as state vector the duration of the services $\Delta^k = (\Delta_1^k, \Delta_2^k, \ldots, \Delta_n^k)^t$, which results in (see [6])

$$\Delta^{k+1} = J\Delta^k + (1-\rho)^{-1}\|\delta\|_1(I-J)\vec{\rho}. \tag{2}$$

Another form of expressing the trajectory is the mixed dynamics, proposed by [8], given by

$$y^{k+1} = y^k + \Delta^k - (1-\rho)^{-1}\|\delta\|_1\vec{\rho}, \tag{3}$$

$$\Delta_i^k = \frac{1}{M_{ii}}\left(b_i - M_i y^{k-1} - \sum_{j=1}^{i-1} M_{ij}\Delta_j^k + \sum_{j=1}^{i} s_j M_i\vec{\rho}\right). \tag{4}$$

Each of these descriptions has an application on the results that will follow.

One of the main points to be noted is that the definition of stability is clearly associated to the matrix $M = L + U = AV$, more precisely, by the characteristics of the matrix $J = -L^{-1}U$.

An important property of J is that

$$M \in \mathcal{S} \ \Rightarrow\ J = -L^{-1}U \geqq \mathbf{O}, \text{ i.e., } \forall(i,j), J_{ij} \geqq 0.$$

This result is a direct application of lemma 1 of [7].

Moreover, this implies that stability (i.e., boundness on $|x_j(t)|$, $j = 1, 2, \ldots, n$) occurs if and only if the spectral radius of J is smaller than one ($\lambda(J) < 1$), as can be shown using results of Frobenius-Perron eigenvalues.

In [8], using the mixed dynamics approach it was show that a necessary and sufficient condition for stability is that

$$C(M, \vec{0}) = \{y | My \leqq \vec{0}\} \subset \mathbb{R}_-^n$$

and this is equivalent to $M^{-1} \geqq \mathbf{O}$, that can be tested in polynomial time.

Under these conditions, it has been shown ([8])

Theorem 3.1 *(Stability Theorem) Given any system, obeying the capacity condition ($\rho < 1$), operating under the corridor policy defined by the cone $C(A, b)$, with $M = AV \in \mathcal{S}$*

$$M^{-1} \geqq \mathbf{O} \Leftrightarrow \begin{cases} (i) & \textit{the system is stable;} \\ (ii) & \textit{its trajectory converges to a limit cycle;} \\ (iii) & \textit{the limit cycle is unique.} \end{cases}$$

■

Moreover in [8], it has been shown that assuming M non-singular and $M \in \mathcal{S}$, then

$$\left\{\exists y > 0 \text{ such that } My \geqq \vec{0}\right\} \Leftrightarrow M^{-1} \geqq \mathbf{O}.$$

We shall assume that $M \in \mathcal{S}$, *$M(A)$ is non-singular* and that all the products are subject to demand ($d > \vec{0}$). This implies that $\vec{\rho} > \vec{0}$, and so $\left\{Ad = (1-\rho)^{-1}M\vec{\rho} \geqq \vec{0}\right\}$ *will insure stability.*

We shall call matrices that obey the above, *stable matrices* and refer to *stable A* or *stable M*, to facilitate the remaining of the exposition.

An important result that can be derived using the above theorem and equation (2) is a generalization of the finite reachability lemma presented in [8]

Theorem 3.2 *(Finite Reachability Theorem) Given any system obeying the capacity condition operating under the corridor policy defined by $C(A,b)$, with $M = AV \in \mathcal{S}$ and M lower triangular ($M = L$),*

(i) the system is stable;

(ii) the trajectory after the first round-robin sequence is equal to the limit cycle, i.e., $y^2 = y^3 = \ldots = y^*$ ($\forall k > 1, y^k = y^*$);

(iii) the time needed to reach the limit cycle is given by $\|[y^* - (y^1 - (1-\rho)^{-1}\|\delta\|_1\vec{\rho})]\|_1 + \|\delta\|_1$, when y^1 is the initial state in the V-basis.

Proof

(i) Using lemma 4 in [7], as $L = M \in \mathcal{S}$ and L lower triangular, $M^{-1} = L^{-1} \succeqq \mathbf{O}$, and the stability is guaranteed by theorem 3.1.

As $J = -L^{-1}U = \mathbf{O}$, equation (1), $\forall k \in I\!N$, $y^{k+1} = L^{-1}b + L^{-1}SL\vec{\rho} - (1-\rho)^{-1}\|\delta\|_1\vec{\rho} = y^*$.

(ii) A simple inspection of equations (2) and (1) will indicate that $y^{k+1} = y^*$, for all $k > 1$, implies $\Delta^{k+1} = \Delta^*$, for all $k > 1$. I.e., we reached a cycle.

Using the theorem 3.1, $M^{-1} \succeqq \mathbf{O}$ implies in convergence of the trajectory to an unique limit cycle. So, clearly, this cycle is the limit cycle.

(iii) The time needed to reach the limit cycle is given by $t^2 = \sum_{j=1}^{n} \Delta_j^1 + \sum_{j=1}^{n} \delta_j$, and equation (3) states that $y^* = y^1 + \Delta^1 - (1-\rho)^{-1}\|\delta\|_1\vec{\rho}$, so,

$$t^2 = \|\Delta^1\|_1 + (1-\rho)^{-1}\|\delta\|_1 = \left\|y^* - [y^1 - (1-\rho)^{-1}\|\delta\|_1\vec{\rho}]\right\|_1 + \|\delta\|_1.$$

∎

4 Optimal Design of Corridor Policies in an Infinite Horizon

We shall present our results considering as cost to be minimize the functional $\mathcal{F}$ given by

$$\mathcal{F} = limsup_{t\to\infty}\frac{1}{t}\sum_{j=1}^{n}\int_0^t (c_j^+ x_j^+(s) + c_j^- x_j^-(s))ds$$

$$\text{where} \begin{cases} x^+ &= max\{0,x\} \\ x^- &= min\{0,x\} \\ c_j^+ &= \text{unit stock cost for product } j \geqq 0 \\ c_j^- &= \text{unit penalty for unfullfilled demand of product } j \geqq 0. \end{cases}$$

This cost has been introduced elsewhere ([1], [2], [5], [4]) and we shall accept it as it portraits a relevant situation. Moreover, it is the kind of cost commonly used to justify the the hierarchical control using tracking, that is the inspiration for corridor policies ([3]).

Due to theorem (3.1) and the fact that if M (or equivalently A) is not stable, then the trajectory leads to $\mathcal{F} = +\infty$, we shall concentrate our analysis just in the behaviour of the limit cycle. For this we shall use the following facts:

Lemma 4.1 *If the trajectory converges to a limit cycle then the duration of each production run in the cycle is independent of the cone $C(A,b)$.*

Proof The convergence for a limit cycle happens if and only if the spectral radius of J is smaller than 1 and so $(I - J)$ is non-singular. Then, equation (2) becomes

$$\overline{\Delta} = (1-\rho)^{-1}\|\delta\|_1\vec{\rho},$$

when $\overline{\Delta}$ is vector of production run durations in the limit cycle. ■

This result can be also shown by elementary linear algebra ([3]), as $\sum_{i=1}^{n}\overline{\Delta}_i v^i + \|\delta\|_1 v^0 = 0$ in the limit cycle and V is non-singular.

An immediate corollary of lemma 4.1 is that the duration T of a limit cycle independs of $C(A,b)$ $(T = \frac{\|\delta\|_1}{1-\rho})$. The same independence is not true with respect of the actual trajectory in a limit cycle, as by equation (1) we have

$$\begin{aligned} \overline{y} &= M^{-1}(b + SM\vec{\rho} - (1-\rho)^{-1}\|\delta\|_1 L\vec{\rho}) && (5) \\ \overline{x} &= A^{-1}(b + SM\vec{\rho} + (1-\rho)^{-1}\|\delta\|_1 L\vec{\rho}) && (6) \end{aligned}$$

where $\overline{y}$ $(\overline{x})$ is the state when the set-up time for task 1 is initiated in the limit cycle associated to $C(A,b)$.

The above implies that given a limit cycle trajectory (coherent with the duration depicted in lemma 4.1), this trajectory is completely specified by the knowledge of $\overline{y}$. The above formulas indicate also that if we know the desired trajectory, given any stable M (stable A), we can reach such a trajectory with a convenient choice of b.

Another way of characterizing the trajectory of the limit cycle is the knowledge of the minimum values of the excess production for product j (x_j^{min}) for $j = 1, 2, \ldots, n$, along this given limit cycle:

Lemma 4.2 *(Separability Lemma) If $x(.)$ is a trajectory associated to the limit cycle of a stable corridor policy induced by the cone $C(A,b)$, $x(.)$ is completely characterized by the minimum values of each $x_j(.)$ $(j = 1, 2, \ldots, n)$ along this cycle, denoted by x_j^{min}. Moreover, given any stable matrix $\overline{A}$ (respectively $\overline{M}$), there is a $\overline{b}$ such that this cycle is the limit cycle associated to $C(\overline{A}, \overline{b})$.*

Proof The proof is trivial, verifying that the balance equation

$$\frac{1}{\tau_j}\overline{\Delta}_j = d_j T$$

and the definition of x_j^{min}, $j = 1, 2, \ldots, n$, leads to the unique determination of $\overline{x}$. Applying equation (6) with the observation that any stable matrix is non-singular, the second part of the lemma follows, as the conditions $\overline{A}d \geqq \vec{0}$ and $\overline{M} \in \mathcal{S}$ imply that $x(.)$ is containned in the cone $C(\overline{A}, \overline{b})$. ■

With respect to design procedures, the above result has the fundamental implication that for any *separable average criterion*, the determination of the optimal corridor policy becomes a separable problem and each subproblem has exactly one variable. In our case, defining

$$\mathcal{F}_j(x(.)) = \mathcal{F}_j(x_j^{min}) = \int_0^T (c^+ x_j^+(s) + c_j^- x_j^-(s))ds,$$

where $x_j(.)$ is the limit cycle trajectory for task j defined by x_j^{min} $(j = 1, 2, \ldots, n)$, we have

$$\begin{aligned} \mathcal{F}^* &= inf_{(A,b)} \lim\sup_{t\to\infty} \tfrac{1}{t} \textstyle\int_0^t (c^+ x_j^+(s) + c_j^- x_j^-(s))ds = \\ &= \tfrac{1}{T} \inf_{x_1^{min}, x_2^{min}, \ldots, x_n^{min}} \textstyle\int_0^T (c^+ x_j^+(s) + c_j^- x_j^-(s))ds = \\ &= \tfrac{1}{T} \textstyle\sum_{j=1}^n \inf_{x_j^{min}} \mathcal{F}_j(x_j^{min}) = \tfrac{1}{T} \sum_{j=1}^n \mathcal{F}_j^*. \end{aligned}$$

For the cost here considered, the determination of $\mathcal{F}_j^*$ and of $x_j^* \in \arg\min \mathcal{F}_j(x_j^{min})$ is quite simple. Denoting $x_j^{max} = max_{t\in[0,T]} x(t)$, it is trivial to show that $x_j^{max} \geqq 0$ and $x_j^{min} \leqq 0$ for any candidate for minimizer of $\mathcal{F}_j(x_j^{min})$.

Using this and the condition $x(\vec{0}) = \vec{0}$, the trajectory of each excess production can be drawn as

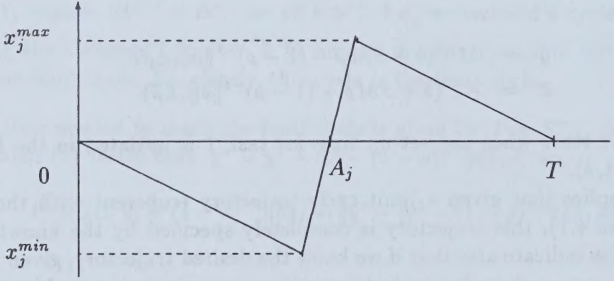


Figure: excess production trajectory for product j in the limit cycle

where the inclination of the doward lines is $-d_j$, the inclination of the upward line is $(\frac{1}{\tau_j} - d_j)$ and A_j is defined as $\min\{t \in (0, T] | x_j(t) = 0\}$.

Some elementary area calculations lead to

$$\begin{aligned} \mathcal{F}_j^* &= \tfrac{1}{2} d_j (1 - \rho_j) \tfrac{c_j^- c_j^+}{c_j^- + c_j^+} T^2, \\ A_j^* &= \tfrac{c_j^+}{c_j^- + c_j^+} T \\ x_j^* &= -d_j (1 - \rho_j) \tfrac{c_j^+}{c_j^- + c_j^+} T \end{aligned}$$

This leads to the following theorem,

Theorem 4.1 *Defining*

$$\overline{x}_k = x_k^* + \sum_{j=1}^{k} \delta_j + \sum_{j=1}^{k-1} \overline{\Delta}_j$$

and

$$b^*(A) = A\overline{x} - SM\vec{\rho} - (1-\rho)^{-1}\|\delta\|_1 L\vec{\rho},$$

the optimal corridor policies associated to the minimization of $\mathcal{F}$ are defined by

$$\{(A,b)|A \textit{ is stable and } b = b^*(A)\}$$

leading to the optimal value

$$\mathcal{F}^* = \frac{T^2}{2}\sum_{j=1}^{n} \frac{c_j^- c_j^+}{c_j^- + c_j^+} d_j(1-\rho_j).$$

Proof The proof is omitted as it is a trivial consequence of the previous lemmas, and preceding comments and equation (6). ■

We must emphasize that the above methodology applies to all separable average criteria.

5 Final comments

The results depicted in the previous sections lead to some interesting constations and some additional results, here briefly sketched for presentation reasons.

A first non-intuitive result is that for any stable A matrix there is a $b^*(A)$, such that $(A, b^*(A))$ is an optimal corridor policy. This result contradicts the intuition that a good solution would be associated to a narrow tracking. Quite on the contrary, the optimal corridor policy does not depend on the solution of the high-level planning problem!

Another non-intuitive result is that for trajectories in the relaxed problem, that have a target point, the vertex of an optimal corridor, in general, is not the target point. This can be seen by the trivial example where $c_j^+ = 0$, for all $j = 1, 2, \ldots, n$, where the target point obtained by the relaxed problem is obviously the origin. Moreover, $c_j^+ = 0$ also implies, $x_j^* \geqq 0$, leading to $b(A) \geqq (y_1^*, y_2^*, \ldots, y_n^*) > \vec{0}$.

With respect to transient analysis, the intuition is to eliminate transients, using the Finite Reachability Theorem (3.2), i.e., choosing M to be lower triangular with $M \in \mathcal{S}$. This can be shown to be true under some technical conditions (for instance, x^0 large enough to imply $\Delta_j^k \geqq \overline{\Delta}_j$, for all (k,j)). At this point, we defend the adoption of lower triangular M matrices using the argument of operational simplicity.

Moreover, for a given system $\mathcal{F}_j^*$ is proportional to the period T, so the optimal round-robin sequencing is the one associated to the minimization of $\|\delta\|_1 = \sum_{j=1}^n \delta_j$. In general terms, this corresponds to the solution of a traveling salesman problem, but in actual applications this question may have a simple solution.

Although this work present some results with respect to the optimization of linear corridor policies (under separable average criteria), a number of questions has yet to be solved: operation under failure, a more comprehensive transient analysis, discounted costs ($\limsup_{t\to\infty} \int_0^t e^{-\alpha s} cost(s)ds$), ... It is our hope that this work will lead to other results and, mainly, to the acceptance that linear corridor policies are scheduling policies with limit cycle independent of the solution of a high level planning problem.

6 Bibliography

[1] J.Kimenia and S.B.Gershwin, 1983. *An Algorithm for the Computer Control of Flexible Manufacturing Systems*, IIE Transactions, Vol.15, No.4, 353-362, dezember.

[2] R. Akella and P. R. Kumar, 1986. *Optimal control ofproduction rate in a failure prone manufacturing system*, IEEE Trans. on Automatic Control, Vol. AC-31, 116-126, february.

[3] A. Sharifnia, M. Caramanis e S. Gershwin, 1991. *Dynamical setup scheduling and Flow control in Manufacturing Systems.* Discrete Event Dynamic Systems: Theory and Applications, 1:149-175.

[4] A.M. Geoffrion, *Elements of Large Scale Mathematical Programming*, Management Science, Vol. 16, No. 11, 652-691, july.

[5] A. Sharifnia, 1994. *Stability and Performance of Distributed Production Control Methods based on Continuous Flow Models*, IEEE Trans. on Automatic Control, Vol. 39, No. 4, 725-737, april.

[6] L. O. Brandão, 1994. *Políticas Hipercone para Sistemas de Manufatura Flexíveis*, master thesis, Instituto de Matemática e Estatística - USP, Brazil, march.

[7] J. R. Perkins and C.Humes Jr. and P. R. Kumar, 1994. *Distributed control of flexible manufacturing systems: stability and performance*, IEEE Transactions on Robotics and Autonomation, Vol. 37, No. 1, 132-141, april.

[8] C.Humes Jr., L.O.Brandão and M.P.Garcia, 1995. *A Mixed Dynamics Aproach for Linear Corridor Policies (A Revisitation of Dynamic Setup Scheduling and Flow Control in Manufacturing Systems)*, "Discrete Event Dynamic Systems: Theory and Aplications", No. 5, 59-82, january.

[9] Guo-Xiang Yu and P. Vakili. *Periodic and Chaotic Dynamics of a Switched Server System under Corridor Policies*, pre-print.

[10] M.P. Garcia and L.O. Brandão. *Coerência de Políticas de Corredores Lineares*, pre-print.

PLC Implementation of a DES Supervisor for a Manufacturing Testbed

R. J. Leduc and W. M. Wonham

Systems Control Group
Department of Electrical and Computer Engineering
University of Toronto
Toronto, Ont., M5S 1A4
email: leduc@control.toronto.edu, wonham@control.toronto.edu

Abstract - We describe an implementation of Ramadge-Wonham (RW) discrete-event supervisors for a manufacturing testbed. Testbed modeling and supervisor design are discussed. Modular control architecture is adopted. Component supervisors are represented as equivalent clocked Moore synchronous state machines (CMSSM), which finally are implemented on PLCs.

1 Introduction

In this paper we address the conversion of a theoretical discrete event system (DES) supervisor, designed using RW supervisory control theory ([8], [12]), to a physical implementation on programmable logic controllers (PLCs:[6], [9]). This has been shown to be feasible by Brandin ([2], [3]) for specific applications; here we propose a general approach. A PLC based testbed that physically simulates a manufacturing workcell was built for illustration.

We describe the modeling of the testbed, and the design of its controllers. An implementation method for DES supervisors using clocked Moore synchronous state machines (CMSSM:[11]) is proposed. A formal model for CMSSM is defined and it is shown that a CMSSM contains equivalent control information. Finally, implementation of a CMSSM on a PLC is discussed.

2 Description of Testbed

The testbed is designed to simulate a manufacturing workcell, in particular, problems of routing and collision. The configuration of the testbed (Fig. 1) is similar to a portion of the product routing design of the Motorola Fusion Factory [10].

The testbed is composed primarily of model railroad components. The tracks are laid out to resemble a set of three interacting work units. The two trains simulate Automated Guided Vehicles (AGVs) that convey material to and from the manufacturing units. Each unit has a small electric crane to simulate robots loading and unloading the AGVs.

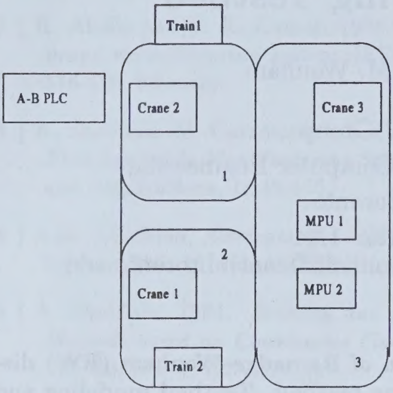

Train1
A-B PLC
Crane 2
Crane 3
1
MPU 1
2
Crane 1
MPU 2
Train 2
3

Figure 1: Layout of Testbed

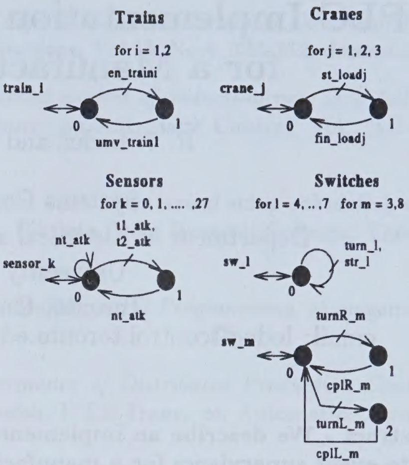

Trains
for i = 1,2
en_traini
train_i
0
1
umv_train1
Cranes
for j = 1, 2, 3
st_loadj
crane_j
0
1
fin_loadj
Sensors
for k = 0, 1, ... ,27
t1_atk,
t2_atk
nt_atk
sensor_k
0
1
nt_atk
Switches
for l = 4, .. ,7 for m = 3,8
turn_l,
str_l
sw_l
0
turnR_m
sw_m
0
1
cplR_m
turnL_m
2
cplL_m

Figure 2: Fundamental Plant Models

The system contains six remote track switches to control the paths of the trains, plus 26 sensors that detect the presence and identity of each train.

The testbed is controlled by two embedded MC68332 processors and an Allen-Bradley PLC. The microprocessors control the trains and cranes directly, and also have a discrete interface for communication with the PLC. The PLC is responsible for control of the overall system.

3 DES Models and Supervisors

In this section we discuss the models used for the testbed, the desired control specifications, and the supervisors that enforce these specifications.

3.1 Plant Model

The plant model for the testbed consists of the following basic elements: trains, cranes, sensors and switches. They are shown in Fig. 2. These models are augmented by additional specifications that define how the elements interact with one another. In total, there are 108 modular specifications.

The interaction models show the interdependencies of the basic models. The first series of models (Fig. 3) shows how the sensors depend on train movement. If all trains are stationary, then every sensor is disabled. The next series of models represents the sensors' interdependencies. With respect to the starting position of a particular train, sensors can only be reached in a particular order, dictated by their physical location on the testbed.

The following series of models detail the sensors' dependency on the switches. Depending on the current location of the trains, certain sensors are inaccessible if the switch prevents the train's approach. An example of this model is given in Fig. 4.

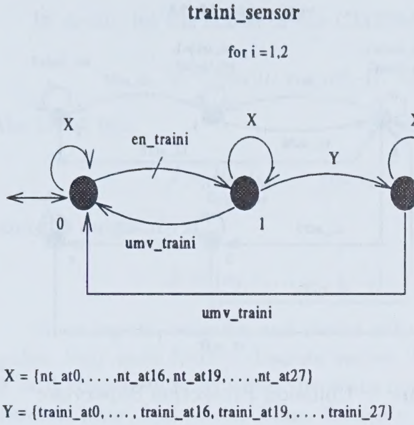

traini_sensor
for i = 1,2
X
en_traini
X
X
Y
0
1
2
umv_traini
umv_traini
X = {nt_at0, . . . ,nt_at16, nt_at19, . . . ,nt_at27}
Y = {traini_at0, . . . , traini_at16, traini_at19, . . . , traini_27}

Figure 3: Sensor Dependencies on Trains

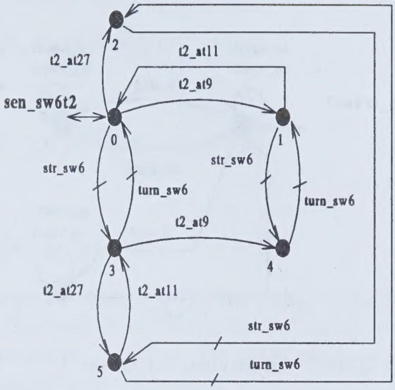

2
t2_at27
t2_at11
t2_at9
sen_sw6t2
0
1
str_sw6
str_sw6
turn_sw6
turn_sw6
t2_at9
3
4
t2_at27
t2_at11
str_sw6
5
turn_sw6

Figure 4: Sensor Dependency for Switch 6

The final series of models is slightly different from the previous series. This series, the switch request handlers, manages how the controllers interact with the switches. The handlers disable the switches' activation events until they are specifically requested.

3.2 Control Specifications

The testbed's safety and operation specifications are as follows: prevent trains from colliding, ensure switches are set correctly while a train is traversing them, enforce specified routes, and synchronize trains to permit loading by cranes.

3.3 DES Supervisors

After the plant model was developed, 29 modular supervisors were designed to enforce the given control specifications. The controllers work as follows. The routing controllers track the progress of each train and activate track switches when necessary.

A crane controller monitors its immediate sensor for a train. When a train arrives, it is stopped, loaded and then released. The crane controller for crane 1 is shown in Fig. 5.

The final class of supervisors are the collision protection controllers. Each track section has its own controller that permits only one train to occupy that portion of the track at any time. Fig. 6 shows an example controller. Note that Fig. 5 and Fig. 6 are complete definitions of the indicated supervisors except for the omission of unimportant self-looped events.

3.4 Discussion of Design Results

After the testbed was modeled, it became apparent that centralized control would be infeasible due to the large size of the composite model (on the order of 10^{16} states). By use of Theorem 2 in [4], we were able to verify controllability for the testbed's 29 modular supervisors. The cited theorem identifies a reduced plant model that can be

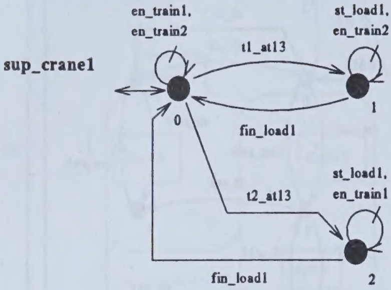

sup_crane1
en_train1,
en_train2
st_load1,
en_train2
t1_at13
0
1
fin_load1
st_load1,
en_train1
t2_at13
fin_load1
2

Figure 5: Supervisor for Crane 1

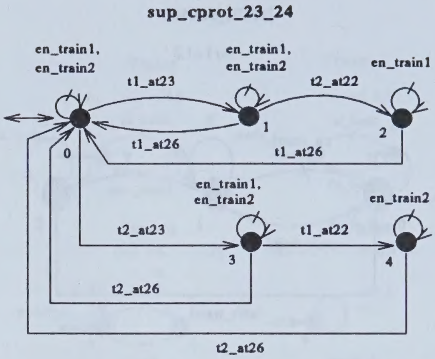

sup_cprot_23_24
en_train1,
en_train2
en_train1,
en_train2
en_train1
t1_at23
t2_at22
0
1
2
t1_at26
t1_at26
en_train1,
en_train2
en_train2
t2_at23
t1_at22
3
4
t2_at26
t2_at26

Figure 6: Collision Protection Supervisor

used to verify controllability for a given supervisor. Nonblocking of the testbed has not yet been verified computationally owing to the strong interaction among its component sub-plants.

4 PLC Implementation

Our proposed implementation technique is to translate supervisors into CMSSM, and then to express the boolean logic defining a CMSSM in Relay Ladder Logic for execution on the PLC.

4.1 Synchronous State Machines

A clocked synchronous state machine is a standard way to implement control functions in digital hardware. The state machine only changes states on the rising (falling) edge of a clock pulse. A Moore machine is such that the output is a function only of the present state.

In a CMSSM, all variables represent a binary digit, and the next state vector is determined by a boolean logic equation in the input variables and the current state variables.

4.2 Formal Model

To prove that a CMSSM is an equivalent control representation, we first present a formal model.

Let the CMSSM be represented by the 7-tuple

$$H = (I, Z, Q, \Omega, \Phi, Q_{Res}, T)$$

Here I represents inputs to the state machine; Z represents outputs from the state machine; Q represents the state vector; Ω is the next state function; Φ is the state to output map; Q_{Res} represents the initial or reset state, and T is the period of the clock pulse that drives the state machine.

In detail, let the states of the CMSSM be represented by:

$$Q = [q_0, q_1, \dots, q_{l-1}], \; q_j \in \{0,1\}, \; j = 0, 1, \dots, l-1;$$

the input by:

$$I = [i_0, i_1, \dots, i_{v-1}], \; i_j \in \{0,1\}, \; j = 0, 1, \dots, v-1;$$

and the output by:

$$Z = [z_0, z_1, \dots, z_{r-1}], \; z_j \in \{0,1\}, \; j = 0, 1, \dots, r-1$$

Since inputs, outputs, and states only change on the rising (falling) edge of the clock pulse, they each form a discrete series. Let k represent the index for the series, where $k = 0$ corresponds to when the state machine starts, k = 1 to the state machine $1 \cdot T$ seconds later, and so on. Thus, the state variables form a series:

$$Q(k), \; k = 0, 1, 2, 3, \dots \quad \text{where } Q(k) = [q_0(k), q_1(k), \dots, q_{l-1}(k)].$$

with $Q(0) = Q_{Res}$. The input and output vectors form similar series.

The new input, $I(k+1)$ is simply the sampled value of the input at the $(k+1)^{th}$ clock pulse. The CMSSM doesn't model how the input changes, but simply makes decisions based on its last state and the new input.

The next value of Q is determined by the next state function, Ω, defined as follows:

$$\Omega : Q \mathrm{x} I \to Q; \quad Q(k+1) = \Omega(Q(k), I(k+1))$$

The current output is determined by the output map, Φ, defined as follows:

$$\Phi : Q \to Z; \quad Z(k+1) = \Phi(Q(k+1))$$

4.3 Translation Method

In this section, we define a translation method to convert a deterministic DES supervisor into a CMSSM. We present both a centralized and modular implementation. First, we state a few definitions.

Let the plant model be defined as: $G = (Y, \Sigma, \eta, Y_0, Y_m), \; \Sigma = \Sigma_u \cup \Sigma_c$
Let $S_1, S_2, \dots, S_n$ be n supervisors for G, with

$$S_j = (X_j, \Sigma, \xi_j, X_{0_j}, X_{m_j}), \; j = 1, 2, \dots n$$

Let $S = \text{meet}(S_1, S_2, \dots, S_n) = (X, \Sigma, \xi, X_0, X_m)$

4.3.1 Centralized Implementation

For the centralized implementation, the CMSSM will be defined with respect to the composite supervisor, S.

Define the state size of Q to satisfy $2^l \geq \mid X \mid$. Define an arbitrary injective map, $\lambda : X \to Q$. The initial state is defined to be $Q_{Res} = \lambda(X_0)$.

Define the state size of I to be $v = \mid \Sigma \mid$. Define an arbitrary bijective map, $\gamma : \Sigma \to \{i_0, i_1, \dots, i_{v-1}\}$. Then define an injective map, $\Gamma : \Sigma \to I$ where:

$$\Gamma(\sigma) = [0 \cdots 0 \ \underset{j}{1} \ 0 \cdots 0] \text{ for } \sigma \in \Sigma \text{ and } \gamma(\sigma) = i_j$$

Define the state size of Z to be $r = | \ \Sigma_c \ |$. Define an arbitrary bijective map, $\delta : \Sigma_c \rightarrow \{z_0, z_1, \ \ldots \ , z_{r-1}\}$.

The interpretation of the input variables (i_0, i_1 etc.) is that the CMSSM senses the occurrence of an event σ when its corresponding input $\gamma(\sigma)$ is true (=1). The interpretation of the output variables (z_0, z_1 etc.) is that a controllable event, $\sigma \in \Sigma_c$, is enabled when its corresponding output $\delta(\sigma)$ is true. Otherwise, the event is disabled.

The next state function, Ω, is defined in accordance with the supervisor. For $x \in X$ and $\sigma \in \Sigma$,

$$\text{if } \xi(x,\sigma)! \text{ then } \quad \Omega(\lambda(x), \Gamma(\sigma)) = \lambda(\xi(x,\sigma)) \text{ else } \Omega(\lambda(x), \Gamma(\sigma)) = \lambda(x)$$

All remaining values of Ω are assigned arbitrarily.

The output map, Φ, is similarly based on the supervisor. For $x \in X$ and $\sigma \in \Sigma_c$,

$$\text{if } \ \xi(x,\sigma)! \ \text{ then } \ \delta(\sigma) = 1 \text{ for the state } \lambda(x). \text{ Thus, } \Phi(\lambda(x)) = [\cdots \underset{j}{1} \cdots] \text{ for } \delta(\sigma) = z_j$$

$$\text{otherwise } \ \delta(\sigma) = 0 \text{ and } \Phi(\lambda(x)) = [\cdots \underset{j}{0} \cdots] \text{ for } \delta(\sigma) = z_j$$

4.3.2 Modular Implementation

For simplicity, we define the modular implementation for the case of two modular DES supervisors, S_1 and S_2. The definition can easily be extended to n supervisors.

First, the two supervisors are converted to centralized CMSSM as above. But, instead of actually controlling the system's outputs, they generate 'local' outputs. Thus we have the following controllers:

$$H_1 = (I, Z_1, Q_1, \Omega_1, \Phi_1, Q_{Res_1}, T) \quad \text{and} \quad H_2 = (I, Z_2, Q_2, \Omega_2, \Phi_2, Q_{Res_2}, T)$$

The Composite Controller, H_{mod}, is then implemented by taking the conjunction of the two modular controllers' outputs.

We then define the following maps:

i) $\zeta = \zeta_1 \wedge \zeta_2 \ \ (X_1 \text{ x } X_2) \ = \ \zeta_1(X_1) \wedge \zeta_2(X_2)$
ii) $\Omega = \Omega_1 \text{ x } \Omega_2 \ (Q_1 \text{ x } Q_2) \ = \ \Omega_1(Q_1) \text{ x } \Omega_2(Q_2)$
iii) $\Phi = \Phi_1 \wedge \Phi_2 \ (Q_1 \text{ x } Q_2) \ = \ \Phi_1(Q_1) \wedge \Phi_2(Q_2)$

Thus, the controller is defined: $H_{mod} = (I, Z_1 \wedge Z_2, Q_1 \text{x} Q_2, \Omega, \Phi, Q_{Res_1} \text{ x } Q_{Res_2}, T)$

4.4 Control Equivalence

For our purposes, we are only interested in whether a CMSSM will enforce the same control action as the original DES supervisor. The effects of transmission delay and concurrent events are not considered. For a discussion on these subjects, refer to [1], [5] and [7]. To aid in the following discussion, some definitions are useful.

Definition: A DES supervisor and a CMSSM are control equivalent if they both take the same control action based on any sequence of events generated by the plant.

Define $\zeta : X \rightarrow Z$, to map the control action specified in the supervisor at a given state to the same representation used by the CMSSM.

4.4.1 Centralized Equivalence

For centralized equivalence we have the following:

Theorem 1 *If S is controllable for plant G then the centralized implementation, H, is control equivalent to S.*

To prove Theorem 1, it is sufficient to show that Fig. 7 commutes. It is clear from the definitions of λ, Γ, Ω, Φ, and ζ that this is in fact the case.

$$\begin{tikzcd}
X \arrow[dd, "\lambda"'] & \times & \Sigma \arrow[dd, "\Gamma"] \arrow[rr, "\xi"] & & X \arrow[dr, "\zeta"] & \\
 & & & & & Z \\
Q & \times & I \arrow[rr, "\Omega"] & & Q \arrow[ur, "\Phi"'] &
\end{tikzcd}$$

$$\begin{tikzcd}
X_1 \arrow[dd, "\lambda_1"'] & \times & X_2 \arrow[dd, "\lambda_2"] & \times & \Sigma \arrow[dd, "\Gamma"] \arrow[rr, "\xi"] & & X_1 \times X_2 \arrow[dr, "\zeta"] & \\
 & & & & & & & Z \\
Q_1 & \times & Q_2 & \times & I \arrow[rr, "\Omega"] & & Q_1 \times Q_2 \arrow[ur, "\Phi"'] &
\end{tikzcd}$$

Figure 7: Control Equivalence Diagram

Figure 8: Modular Control Equivalence Diagram

4.4.2 Modular Equivalence

For modular equivalence, we present the following theorem:

Theorem 2 *If S_1 and S_2 are controllable for plant G then the modular implementation, H_{mod}, is control equivalent to $S_1 \wedge S_2$.*

To prove Theorem 2, it is sufficient to show that Fig. 8 commutes, namely whenever the upper path through the diagram is defined, then so is the lower, and yields the same results. This follows directly from the definitions of the relevant maps. This result can easily be extended to n supervisors.

4.5 Implementation Details

In this section we discuss the implementation of the synchronous state machine's mapping functions, and the execution algorithm used to implement the CMSSM.

For the state machine, both the next state function (Ω) and the output map (Φ) are implemented in a modular manner. The next state function is implemented as l boolean logic functions. Thus,

$$\Omega(Q(k), I(k+1)) = [\omega_0(Q(k), I(k+1)), \ldots, \omega_{l-1}(Q(k), I(k+1))]$$

Similarly, the output map is implemented as r boolean logic functions. Thus,

$$\Phi(Q(k+1)) = [\phi_0(Q(k+1)), \ldots, \phi_{r-1}(Q(k+1))]$$

The execution algorithm used to implement the CMSSM is as follows:

1. Initialize state machine by setting $Q(0) = Q_{Res}$, $Z(0) = \Phi(Q_{Res})$and $k = 0$.
2. Wait for next clock pulse $(k+1)$. Sample inputs and set $I(k+1)$ to these values.
3. Compute $Q(k+1) = \Omega(Q(k), I(k+1))$ and $Z(k+1) = \Phi(Q(k+1)$.
4. Set $k = k+1$ and then go to step 2.

4.6 PLC Interpretation

A programmable logic controller is similar to a microcomputer, but specialized for control purposes. The PLC was chosen for the physical implementation of the synchronous state machine because of its robustness, and its wide use in industry.

The CMSSM will be implemented on the PLC in the form of a relay ladder logic (RLL) program. A RLL program is made up of several steps called rungs. A rung is composed of a logical evaluation statement and an output assignment statement. If the logic evaluates to true, then the output is energized. The basic components of a rung are shown in Fig. 9.

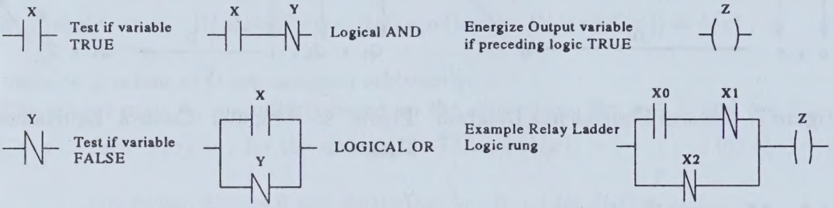

X
Test if variable TRUE
X
Y
Logical AND
Energize Output variable if preceding logic TRUE
Z
Y
Test if variable FALSE
X
Y
LOGICAL OR
Example Relay Ladder Logic rung
X0
X1
Z
X2

Figure 9: Components of Relay Ladder Logic

The execution algorithm given in Section 4.5 will be translated into a RLL program. The code will have five main parts: initialize variables, evaluate next state logic, assign temporary outputs, set new states to be current states, and combine temporary outputs to generate system outputs.

4.7 PLC Controllers

The 29 DES supervisors were implemented using the modular method outlined in Section 4.3.2. As an example, we illustrate the implementation of supervisor *sup_crane1* (Fig. 5). The CMSSM representation is shown in Fig. 10. The reset state, next state function, and the local output map are defined below. Fig. 11 shows the equivalent Relay Ladder Logic for computing the next state of the state variables. The formulas below only show inputs that are relevant to deciding the next state, and only those outputs that the controller actually disables.

$$Qa_{Res} = [0,0]$$

$$\begin{aligned}\omega_1(Qa(k), Ia(k+1)) &= [\neg qa_1(k) \wedge \neg qa_0(k) \wedge \text{t2\_at13(k+1)}] \vee [qa_1(k) \\ &\quad \wedge \neg \text{ f\_ld1(k+1)}] \\ \omega_0(Qa(k), Ia(k+1)) &= [\neg qa_1(k) \wedge \neg qa_0(k) \wedge \text{t1\_at13(k+1)}] \vee [qa_0(k) \\ &\quad \wedge \neg \text{ f\_ld1(k+1)}]\end{aligned}$$

$$\phi_2(Qa(k+1)) = \neg qa_1(k+1) \qquad \phi_1(Qa(k+1)) = \neg qa_0(k+1)$$
$$\phi_0(Qa(k+1)) = qa_1(k+1) \vee qa_0(k+1)$$

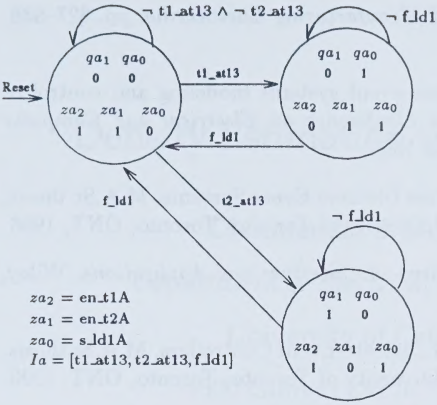

¬ t1_at13 ∧ ¬ t2_at13
¬ f_ld1
Reset
qa_1 qa_0
0 0
za_2 za_1 za_0
1 1 0
t1_at13
f_ld1
qa_1 qa_0
0 1
za_2 za_1 za_0
0 1 1
f_ld1
t2_at13
¬ f_ld1
qa_1 qa_0
1 0
za_2 za_1 za_0
1 0 1
za_2 = en_t1A
za_1 = en_t2A
za_0 = s_ld1A
Ia = [t1_at13, t2_at13, f_ld1]

Figure 10: CMSSM for *sup_crane1*

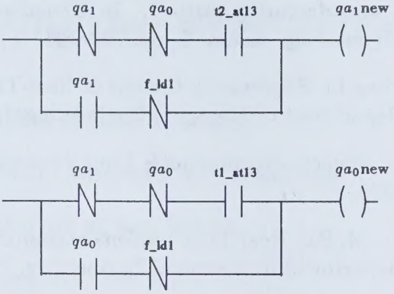

qa_1
qa_0
t2_at13
qa_1new
qa_1
f_ld1
qa_1
qa_0
t1_at13
qa_0new
qa_0
f_ld1

Figure 11: RLL for Next State Equations

4.8 Discussion of Implementation Results

Each DES supervisor has been implemented as a synchronous state machine on the Allen-Bradley PLC. An interrupt-driven, discrete interface has also been implemented on each of the embedded MC68332 processors for communicating with the PLC. All PLC controllers have been tested and are operational.

5 Conclusion

In conclusion, we have designed and built a PLC based testbed for investigating the implementation of DES supervisors. The system was modeled and several RW modular supervisors were designed and verified for controllability.

A method to implement DES supervisors as clocked Moore synchronous state machines has been presented and shown to be an equivalent representation. Also, a method to implement supervisors modularly has been defined and shown to be equivalent to implementing a centralized supervisor.

Finally, all 29 DES supervisors have been translated into CMSSM and implemented on a PLC.

References

[1] S. Balemi. Input/output discrete event processes and communication delays. *Discrete Event Dynamic Systems: Theory and Applications*, 4(1):41–85, 1994.

[2] B. A. Brandin. *Real-Time Supervisory Control of Automated Manufacturing Systems.* Ph.D thesis, Department of Electrical Engineering, University of Toronto, 1993.

[3] B.A. Brandin, W.M. Wonham, and B. Benhabib. Discrete-event systems supervisory control applied to the management of manufacturing workcells. In *7th Inter-*

national Conference on Computer Aided Manufacturing Engineering, pp. 527–536, Cookeville, Elsevier, 1991.

[4] R.J. Leduc and W.M. Wonham. Discrete event systems modeling and control of a manufacturing testbed. In *Canadian Conference on Electrical and Computer Engineering*, volume 2, pp. 793–796, Sept 1995.

[5] Yong Li. Supervisory Control of Real-Time Discrete-Event Systems. M.A.Sc thesis, Department of Electrical Engineering, University of Toronto, Toronto, ONT, 1986.

[6] G. Michel. *Programmable Logic Controllers- Architecture and Applications.* Wiley, 1990.

[7] T.- M. Pai. Real-Time Implementation of Discrete-Event Controllers. M.A.Sc thesis, Department of Electrical Engineering, University of Toronto, Toronto, ONT, 1990.

[8] P. Ramadge and W. Wonham. Supervisory control of a class of discrete-event processes. *SIAM J. Control And Optimization*, 25(1):206–230, 1987.

[9] J. Stenerson. *Fundamentals of Programmable Logic Controllers, Sensors, and Communications.* Prentice Hall, 1993.

[10] R. Strobel and A. Johnson. Pocket pagers in lots of one. *IEEE Spectrum*, pp. 29–32, Sept 1993.

[11] J. Wakerley. *Digital Design Principles.* Prentice-Hall, 1990.

[12] W. Wonham and P. Ramadge. On the supremal controllable sublanguage of a given language. *SIAM J. Control and Optimization*, 25(3):637–659, 1987.

Density Estimation using Neural Networks*

Dharmendra S. Modha and Elias Masry
Department of Electrical and Computer Engineering
University of California at San Diego
9500 Gilman Drive, La Jolla, CA 92093-0407
emails: dmodha@sdcc3.ucsd.edu, masry@ece.ucsd.edu

Abstract

Given N i.i.d. observations $\{X_i\}_{i=1}^N$ taking values in $[-\frac{1}{2}, \frac{1}{2}]^d$, such that p^* denotes their common probability density function, we estimate p^* from an exponential family of densities based on single hidden layer sigmoidal networks using a certain minimum complexity density estimation scheme. Assuming that p^* possesses a certain exponential representation, we establish a rate of convergence, independent of the dimension d, for the expected L_1 distance between the proposed minimum complexity density estimator and the true underlying density p^*.

1 Main Result

Let $\{X_i\}_{i=1}^\infty$ be independent and identically distributed (i.i.d.) random variables taking values in $\mathcal{B} \equiv [-\frac{1}{2}, \frac{1}{2}]^d$. Let $p^* : \Re^d \to \Re^+$ denote their common probability density function (density), which is assumed to exist. Given N observations $\{X_i\}_{i=1}^N$ drawn from $\{X_i\}_{i=1}^\infty$, we estimate the density p^*. Specifically, we propose a certain minimum complexity density estimator, say $\hat{p}_N$, using an exponential family of densities based on neural networks, and provide, in the following theorem, a rate of convergence for

---

*This work was supported by the Office of Naval Research under Grant N00014-90-J-1175.

the expected L_1 distance between the proposed estimator $\hat{p}_N$ and the true underlying density p^*.

Theorem 1.1 (bound on the expected L_1 distance) *Suppose that the logarithm of the density p^* satisfies a certain Fourier transform-type representation (see Assumption 2.1), and that a certain technical condition holds (see Assumption 3.1). Then,*

$$E \int_{\mathcal{B}} |\hat{p}_N(x) - p^*(x)|dx = O\left(\frac{\log N}{N}\right)^{1/4}.$$

The proof utilizes results of Barron (1991, 1994), Barron and Cover (1991), and Stone (1991), and can be found in the full paper Modha and Masry (1995).

Observe that the rate of convergence, that is the exponent of N, in Theorem 1.1 does not depend on the dimension d. In other words, the estimator $\hat{p}_N$ mitigates the curse of dimensionality in estimating the density p^*.

The rest of this paper is organized as follows. In Section 2, we present the class of densities for which Theorem 1.1 holds. For this class of densities it is possible to obtain rates of approximation independent of the dimension d using a certain exponential family of densities based on neural networks, which is presented in Section 3. Finally, in Section 4, we construct the advertized estimator $\hat{p}_N$ using the exponential family of densities based on neural networks presented in Section 3.

2 A Class of Target Densities

For $w = (w_1, \ldots, w_d)$ and $x = (x_1, \ldots, x_d)$ in $\Re^d$, let $w \cdot x = \sum_{i=1}^d w_i x_i$ denote the usual inner product on $\Re^d$. For $w = (w_1, \ldots, w_d) \in \Re^d$, let $||w||_1 = \sum_{i=1}^d |w_i|$ denote a norm on $\Re^d$.

Assumption 2.1 *Assume that there exists a complex valued function $\tilde{f}$ on $\Re^d$ and a corresponding normalizing constant*[1] *$C^* > 0$ such that for $x \in \mathcal{B}$, we have*

$$p^*(x) = C^* \exp\left(\int_{\Re^d} \left(e^{iw\cdot x} - 1\right) \tilde{f}(w)\, dw\right) \tag{2.1}$$

and that $\int_{\Re^d} ||w||_1 |\tilde{f}(w)|\, dw \leq C' < \infty$ for some known $C' > 0$. Set $C = \max\{1, C'\}$.

---

[1] The normalizing constant is selected such that the density p^* integrates to one.

Let $f^* \equiv \ln p^*$ denote the log-density of p^*. Then, for $x \in \mathcal{B}$, (2.1) may be equivalently interpreted as

$$f^*(x) \equiv \ln p^*(x) = \ln C^* + \int_{\Re^d} \left(e^{iw \cdot x} - 1\right) \tilde{f}(w)\, dw.$$

In other words, we assume that the log-density f^* has a inverse Fourier transform-type representation on the set $\mathcal{B}$. Assumption 2.1 on the log-density function f^* is inspired by the assumption in Barron (1994) on the regression function in the context of regression estimation using neural networks.

3 Exponential Family of Densities based on Neural Networks

Let $\phi : \Re \to \Re$ denote a sigmoidal function satisfying the following properties.

Assumption 3.1 (Barron 1994) *Assume that*

1. $\phi(u) \to 1$ *as* $u \to \infty$ *and* $\phi(u) \to 0$ *as* $u \to -\infty$.
2. $|\phi(u)| \leq 1$ *and* $|\phi(u) - \phi(v)| \leq A_1'|u - v|$ *for all* $u, v \in \Re$ *and for some* $A_1' > 0$. *Set* $A_1 = \max\{1, A_1'\}$.
3. $|\phi(u) - 1_{\{u>0\}}| \leq A_2'/|u|^{A_3}$ *for* $u \in \Re$, $u \neq 0$, *and for some* $A_3 > 0$ *and* $A_2' \geq 0$. *Set* $A_2 = \max\{1, A_2'\}$.

Fix $n \geq 1$. We now proceed to define an exponential density based on neural networks that has n hidden units. Let $\gamma_n = n(d+2)$ denote the number of parameters required to parametrize such a density. For $1 \leq i \leq n$, let $a_i \in \Re^d$, let $b_i \in \Re$, and let $c_i \in \Re$. Define a γ_n-dimensional parameter vector θ as

$$\theta = (a_1, a_2, \ldots, a_n;\ b_1, b_2, \ldots, b_n;\ c_1, c_2, \ldots, c_n).$$

Now, first define a neural network with n hidden units $f_{(n,\theta)} : \mathcal{B} \to \Re$ parametrized by θ as follows

$$f_{(n,\theta)}(x) = \sum_{i=1}^{n} c_i\, \phi(a_i \cdot x + b_i), \tag{3.1}$$

and define a real valued normalizing bias weight corresponding to θ as

$$\bar{c}_{(n,\theta)} = -\ln \int_{\mathcal{B}} \exp(f_{(n,\theta)}(x))\, dx. \tag{3.2}$$

We can now use (3.1) and (3.2) to define an exponential density $p_{(n,\theta)} : \mathcal{B} \to \Re$ parametrized by θ as follows

$$p_{(n,\theta)}(x) = \exp\left(\bar{c}_{(n,\theta)} + f_{(n,\theta)}(x)\right). \tag{3.3}$$

Observe that for all $x \in \mathcal{B}$, we have that $p_{(n,\theta)}(x)$ is positive, and that by construction $p_{(n,\theta)}$ integrates to one. Thus, $p_{(n,\theta)}$ is a density on $\mathcal{B}$. Let

$$\varpi_n = 2^{(2A_3+1)/A_3}\, A_2^{1/A_3}\, n^{(A_3+1)/(2A_3)}, \tag{3.4}$$

where A_1, A_2, and A_3 are as in Assumption 3.1, and define

$$\mathcal{S}^{(n)} = \left\{ \theta \in \Re^{\gamma_n} : \sum_{i=1}^{n} |c_i| \leq 2C, \max_{1\leq i \leq n} ||a_i||_1 \leq \varpi_n, \max_{1\leq i\leq n} |b_i| \leq \varpi_n \right\}, \tag{3.5}$$

where C is as in Assumption 2.1. The definition of the set $\mathcal{S}^{(n)}$ has been adapted from Barron (1994). Define the exponential family of densities parametrized by elements of $\mathcal{S}^{(n)}$ as

$$\Gamma^{(n)} = \left\{ p_{(n,\theta)} : \theta \in \mathcal{S}^{(n)} \right\}.$$

In other words, $\Gamma^{(n)}$ denotes an exponential family of densities based on neural networks with n hidden units, such that the parameters of the single hidden layer sigmoidal networks are constrained to take values in the set $\mathcal{S}^{(n)}$. In the next section, we estimate the density p^* using an element of the collection $\cup_{n=1}^{N} \Gamma^{(n)}$.

The concept of using an exponential family of densities based on neural networks was first considered in White (1992), where he established that the exponential family of densities, namely $\cup_{n\geq 1}\Gamma^{(n)}$, can approximate, in L_1 sense, any compactly supported density. However, White did not provide a rate of approximation. The following lemma, which is an important intermediate step in establishing Theorem 1.1, provides a rate of convergence in approximating a density which satisfies Assumption 2.1 using the family of densities $\Gamma^{(n)}$.

Lemma 3.1 (rate of approximation) *Suppose Assumptions 2.1 and 3.1 hold, then there exists an element $\theta_n^\star \in \Gamma^{(n)}$ such that*

$$\int_{\mathcal{B}} p^*(x) \log_2 \frac{p^*(x)}{p_{(n,\theta_n^\star)}(x)} dx = O\left(\frac{1}{n}\right).$$

The proof follows from Barron (1994), and can be found in Modha and Masry (1995). Observe that the exponent of n in Lemma 3.1 does not depend on the dimension d.

4 Minimum Complexity Density Estimation Scheme

For each fixed $1 \leq n \leq N$, we can compute the maximum likelihood estimator based on $\Gamma^{(n)}$ as[2]

$$\hat{\theta}(n, N) = \arg\min_{\theta \in S^{(n)}} \left\{ -\frac{1}{N} \sum_{i=1}^{N} \log p_{(n,\theta)}(X_i) \right\}, \tag{4.1}$$

where the set $S^{(n)}$ is as in (3.5) and $p_{(n,\theta)}$ is as in (3.3). A backpropagation algorithm for minimizing (4.1) was described in Modha and Fainman (1994). Define the density corresponding to the maximum likelihood estimator $\hat{\theta}(n, N)$ as

$$\hat{p}_{(n,N)} = p_{(n,\hat{\theta}(n,N))}. \tag{4.2}$$

Given a sequence of maximum likelihood estimators $\{\hat{p}_{(n,N)}\}_{n=1}^{N}$, how should one formulate an estimator of the true density p^* that achieves the smallest expected L_1 distance? On one hand, the approximation error in estimating the density p^* using $\hat{p}_{(n,N)}$ decreases as n increases. On the other hand, the estimation error in estimating the density p^* using $\hat{p}_{(n,N)}$ increases as n increases. Thus, we must select the number of hidden units n so as to achieve a compromise between the approximation error and the estimation error. Specifically, for a fixed regularization constant $\lambda > 1$, compute

$$\hat{n} = \arg\min_{1 \leq n \leq N} \left\{ -\frac{1}{N} \sum_{i=1}^{N} \log \hat{p}_{(n,N)}(X_i) + \lambda \frac{L_{n,N} + 2\log(n+1)}{N} \right\}, \tag{4.3}$$

where $L_{n,N} \equiv [n(d+2)] \log(4\varpi_n e\sqrt{nN})$, ϖ_n is as in (3.4), and $\hat{p}_{(n,N)}$ is as in (4.2). Roughly speaking, the first term on the right-hand side of (4.3) has the meaning of codelength (or empirical likelihood loss) when encoding the observations $\{X_i\}_{i=1}^{N}$ using the density $\hat{p}_{(n,N)}$, while the second term on the right-hand side of (4.3) has the meaning of the description complexity of the density $\hat{p}_{(n,N)}$. Thus, the estimator $\hat{n}$ permits the use

---

[2] To be technically precise, we should compute the maximum likelihood estimator as

$$\hat{\theta}(n, N) = \arg\min_{\theta \in T_{(n,N)}} \left\{ -\frac{1}{N} \sum_{i=1}^{N} \log p_{(n,\theta)}(X_i) \right\},$$

where the finite set $T_{(n,N)}$ is an $(1/\sqrt{nN})$-net of the compact set $S^{(n)}$ with respect to a certain metric on $\Re^{\gamma_n}$.

of a larger number of hidden units, only if the resulting increase in the description complexity is offset by a matching decrease in the codelength.

We can now define the minimum complexity density estimator $\hat{p}_N$ as

$$\hat{p}_N = \hat{p}_{(\hat{n},N)}.$$

In other words, the estimator $\hat{p}_N$ is the element corresponding to $\hat{n}$ in the sequence $\{\hat{p}_{(n,N)}\}_{n=1}^{N}$. The estimator $\hat{p}_N$ was introduced by Barron and Cover (1991) in an abstract setting; in this paper, we have adapted their framework to the exponential family of densities introduced in Section 3.

References

Barron, A. R. 1991. Complexity regularization. In *Proceedings NATO Advanced Study Institute on Nonparametric Functional Estimation*, G. Roussas, ed., pp. 561-576. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Barron, A. R. 1994. Approximation and estimation bounds for artificial neural networks. *Machine Learning*, **14**, 115-133.

Barron, A. R., and Cover, T. M. 1991. Minimum complexity density estimation. *IEEE Transact. Inform. Theory*, **37**(4), 1034-1054.

Modha, D. S., and Fainman, Y. 1994. A learning law for density estimation. *IEEE Transact. Neural Networks*, **5**(3), 519-523.

Modha, D. S., and Masry, E. 1995. Rate of Convergence in density estimation using neural networks. *Submitted for publication*.

Stone, C. J. 1991. Asymptotics for doubly flexible logspline response models. *Ann. Statist.*, **19**(4), 1832-1854.

White, H. 1992. Parametric statistical estimation with artificial neural networks. In *Mathematical Perspectives on Neural Networks*, P. Smolensky, M. C. Mozer, and D. E. Rumelhart, eds. L. Erlbaum Associates, Hilldale, NJ.

QUALITATIVE ANALYSIS OF COHEN-GROSSBERG NEURAL NETWORKS WITH MULTIPLE DELAYS*

Hui Ye Anthony N. Michel Kaining Wang
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556, U.S.A.

Abstract

It is well known that a class of artificial neural networks with symmetric interconnections and without transmission delays, known as *Cohen-Grossberg neural networks*, possesses global stability (i.e., all trajectories tend to some equilibrium). We demonstrate in the present paper that many of the qualitative properties of Cohen-Grossberg networks will not be affected by the introduction of sufficiently small delays. Specifically, we establish some bound conditions for the time delays under which a given Cohen-Grossberg network with multiple delays is globally stable and possesses the same asymptotically stable equilibria as the corresponding network without delays. An effective method of determining the asymptotic stability of an equilibrium of a Cohen-Grossberg network with multiple delays is also presented.

I. Introduction

Cohen-Grossberg neural networks constitute a class of artificial feedback neural networks whose activation equation are of the form

$$\dot{x}_i = -a_i(x_i)[b_i(x_i) - \sum_{j=1}^{n} t_{ij}s_j(x_j)], \tag{1.1}$$

$i = 1, \cdots, n$. In (1.1) x_i denotes the state variable associated with the i-th neuron, the function a_i represents an *amplification function* and b_i is an arbitrary function, however, we will require that b_i be sufficiently well behaved to keep the solutions of equation (1.1) bounded. The matrix $T \triangleq [t_{ij}]_{nxn}$ is a real and symmetric matrix and represents the neuron interconnections. The real function s_j is a sigmoidal nonlinearity representing the j-th neuron. The neural dynamics model (1.1) has been widely studied (see, e.g., [1]-[8]). One of the useful qualitative properties of (1.1) is that it is globally stable (i.e., every

---

*This work was supported in part by the National Science Foundation under Grant ECS93-19352.

trajectory of (1.1) converges to some equilibrium). It is well known that (1.1) includes the Hopfield neural networks as a special case, a class of neural networks that has been studied widely (see, e.g., [4]-[8]). Hopfield neural networks are described by a system of equations of the form

$$\dot{x}_i = -c_i x_i + \sum_{j=1}^{n} t_{ij} s_j(x_j), \tag{1.2}$$

$i = 1, \cdots, n$, where $c_i \geq 0$, and x_i, t_{ij} and s_j are the same as in (1.1). Since neural networks (1.1) and (1.2) have the potential of serving as associative memories and performing parallel computations, some electronic implementations of these neural networks in VLSI technology have been realized (see, e.g., [9]-[11]). However, in the implementation process of artificial neural networks, time-delays are unavoidably introduced, and it is known that such delays can cause systems to oscillate (see, e.g, [12]-[14]). Therefore, it is important to take time delays into consideration and to investigate the qualitative properties of neural networks of the type (1.1) and (1.2) with delays. A class of Hopfield neural networks with identical delay for each state can be described by equations of the form

$$\dot{x}_i(t) = -c_i x_i(t) + \sum_{j=1}^{n} t_{ij} s_j(x_j(t-\tau)), \tag{1.3}$$

$i = 1, \cdots, n$, where $\tau > 0$ denotes the time delay. System (1.3) has recently been studied by several workers (see [12], [15]-[18]). It is shown in [12], [18] that system (1.3) is globally stable and that it possesses the same set of asymptotically stable equilibria as system (1.2), if the delay τ is less than a certain bound.

In the present paper, we extend the results of [18] to a significantly larger class of systems. Specifically, we consider Cohen-Grossberg neural networks (1.1) endowed with multiple delays, described by equations of the form

$$\dot{x}_i(t) = -a_i(x_i(t))\Big[b_i(x_i(t)) - \sum_{j=1}^{n} t_{ij}^{(0)} s_j(x_j(t)) - \sum_{k=1}^{K}\sum_{j=1}^{n} t_{ij}^{(k)} s_j(x_j(t-\tau_k))\Big], \tag{1.4}$$

$i = 1, \cdots, n$, where the $t_{ij}^{(k)}$'s denote the interconnections which are associated with delay τ_k, τ_k denotes the k-th time delay for $k = 0, 1, \cdots, K$, such that $0 = \tau_0 < \tau_1 < \cdots < \tau_K$, and where the x_i, b_i and s_j are the same as the corresponding quantities in equation (1.1). We will establish bound conditions for the delays under which Cohen-Grossberg neural networks with multiple delays described by equation (1.4) will exhibit similar qualitative properties as the corresponding original Cohen-Grossberg neural networks without delays, described by equation (1.1) (i.e., conditions under which system (1.4) is globally stable and has the same local stability properties as system (1.1)).

II. Notation

Let R denote the set of real numbers and let R^n denote real n-space. If $x \in R^n$, then $x^T = (x_1, \cdots, x_n)$ denotes the transpose of x. Let $R^{n \times m}$ denote the set of $n \times m$ real matrices. If $B = [b_{ij}]_{n\times m} \in R^{n\times m}$, then B^T denotes the transpose of B. For $x \in R^n$, let $\|x\|$ denote the Euclidean vector norm, $\|x\| = (x^T x)^{1/2}$, and for $A \in R^{n\times n}$, let $\|A\|$ denote the norm of A induced by the Euclidean vector norm, i.e., $\|A\| = [\lambda_{max}(A^T A)]^{\frac{1}{2}}$. I denotes the identity $n \times n$ matrix.

Let R^+ denote the set of nonnegative real numbers, i.e., $R^+ = [0, +\infty)$. Let X be a subset of R^n and let Y be a subset of R^m. We denote by $C(X, Y)$ the set of all continuous functions from X to Y, and we denote by $C^k(X, Y)$ the set of all functions from X to Y which have continuous derivatives up to order k. With $\tau > 0$, $x \in C([-\tau, +\infty), R^n)$, and $t > 0$, we define $x_t \in C([-\tau, 0], R^n)$ as $x_t(s) = x(t + s)$ for $s \in [-\tau, 0]$. For any $\phi \in C([-\tau, 0], R^n)$, the norm of ϕ, denoted by $|\phi|$, is defined as $|\phi| = max\{\|\phi(t)\| : t \in [-\tau, 0]\}$.

The system (1.4) is said to be *globally stable* if for any solution $x(t)$, $\lim_{t\to\infty} x(t)$ exists. For the definitions of stability and asymptotic stability of an equilibrium of (1.4), refer to any of several standard texts (see, e.g., [19]).

III. Preliminaries

In the present paper, we assume that the Cohen-Grossberg neural networks (1.1) and (1.4) satisfy the following assumptions.

Assumption A:

i) the function a_i is bounded, positive and continuous;

ii) the function b_i is continuous;

iii) $T \triangleq [t_{ij}]_{n\times n}$ is a symmetric matrix;

iv) $s_j \in C^1(R, R)$ is a sigmoidal function (so that $s_j'(x_j) \triangleq \frac{ds_j}{dx_j}(x_j) > 0$, $\lim_{x_j\to+\infty} s_j(x_j) = 1$, $\lim_{x_j\to-\infty} s_j(x_j) = -1$, and $\lim_{|x_j|\to\infty} s_j'(x_j) = 0$);

v) $\lim_{x_i\to+\infty} b_i(x_i) = +\infty$ and $\lim_{x_i\to-\infty} b_i(x_i) = -\infty$. □

Remark 1: Assumption A is hypothesized in many references dealing with feedback artificial neural networks without delays (see, e.g., [3]).

More generally, we could replace part v of Assumption A by some other hypothesis which ensures the boundedness of solutions of (1.4). For example, it can be shown that parts (e) and (f) of Theorem 1 in [1] ensure the boundedness of the solutions of neural network (1.1) and also of neural network (1.4). □

Fact 1 If Assumption A is satisfied for systems (1.1) and (1.4), then any solution of (1.1) and (1.4) is bounded. □

If we let $x = [x_1, \cdots, x_n]^T \in R^n$, $A(x) = diag\{a_1(x_1), \cdots, a_n(x_n)\} \in R^{n\times n}$, $B(x) = [b_1(x_1), \cdots, b_n(x_n)]^T \in R^n$, $T = [t_{ij}]_{n\times n}$, and $S(x) = [s_1(x_1), \cdots, s_n(x_n)]^T$, then equation (1.1) can be rewritten as

$$\dot{x} = -A(x)[B(x) - TS(x)]. \tag{3.1}$$

When delays are present in the Cohen-Grossberg neural networks, we need to modify

(3.1) as

$$\dot{x}(t) = -A(x(t))\Big[B(x(t)) - T_0 S(x(t)) - \sum_{k=1}^{K} T_k S(x(t-\tau_k))\Big], \tag{3.2}$$

where the T_k make up the interconnections associated with delay τ_k, $k = 0, 1, \cdots, K$, so that $T = T_0 + T_1 + \cdots + T_K$, and $0 = \tau_0 < \tau_1 < \cdots < \tau_K$, and where $A(x)$, $B(x)$, T, and $S(x)$ are the same as in equation (3.1). Clearly, equation (3.2) is equivalent to equation (1.4).

In order to ensure that the Cohen-Grossberg neural networks (3.1) (or equivalently (1.1)) possess global stability, we will require that the set of all equilibria of (3.1) be a discrete set. This requirement can be ensured by almost all choices (in the sense of Lebesgue measure) of the interconnection matrix T (see, [1]). For this reason, we assume throughout this paper that system (3.1) (or equivalently (1.1)) satisfies the following assumption.

Assumption B: For any equilibrium x_e of (3.1) , the matrix $J(x_e)$ is nonsingular, where

$$J(x) = -T + diag\Big\{\frac{b_1'(x_1)}{s_1'(x_1)}, \cdots, \frac{b_n'(x_n)}{s_n'(x_n)}\Big\}, \tag{J}$$

and where $b_i'(x_i) \triangleq \frac{db_i}{dx_i}(x_i)$ for $i = 1, \cdots, n$. □

It can be proved by using Sard's Theorem (see Lemma 3.3 of [7]) that the following result is true.

Lemma 1 For almost all $T \in R^{n \times n}$ (except a set with Lebesgue measure 0), system (3.1) satisfies Assumption B. □

Furthermore, by using the Inverse Function Theorem (see Remark 3.4 of [7]), we can establish the following result.

Lemma 2 When system (3.1) satisfies Assumption B, the set of equilibria of system (3.1) is a discrete set. □

IV. Global Stability Analysis of Cohen-Grossberg Neural Networks with Multiple Delays

In the present section, we address the global stability properties of Cohen-Grossberg neural networks with multiple delays described by the retarded type differential-difference equation (3.2), or equivalently by equation (1.2). To establish the main results of the present section, we require the following properties of system (3.2).

Lemma 3 If system (3.2) satisfies Assumption B, then the set of equilibria of system (3.2) is a discrete set. □

Remark 2 The set of equilibria of system (3.2), which is identical to the set of equilibria of system (3.1), is a discrete set, by Lemma 2. Furthermore, it follows from Lemma 1 that for almost all $T \in R^{n \times n}$ (except on a set of Lebesgue measure 0), system (3.2) satisfies Assumption B. □

We are now in a position to establish the main result of this section.

Theorem 1 Suppose that for system (3.2), Assumptions A and B are satisfied, and suppose that

$$\sum_{k=1}^{K}(\tau_k\beta\|T_k\|) < 1, \tag{4.1}$$

where $\beta = \max_{x\in R^n}\|A(x)S'(x)\|$, and $S'(x) \triangleq diag\{s'_1(x_1), \cdots, s'_n(x_n)\}$. Then system (3.2) is *globally stable.*

Proof: Since inequality (4.1) is satisfied, there must exist a sequence of positive numbers $(\alpha_1, \cdots, \alpha_K)$, such that

$$\sum_{k=1}^{K}\alpha_k = 1 \quad \text{and} \quad \tau_k\beta\|T_k\| < \alpha_k, \tag{4.2}$$

for $k = 1, \cdots, K$.

To prove the present result, we define for any $x_t \in C([-\tau_K, 0], R^n)$ an energy functional $E(x_t)$ associated with (3.2) by

$$\begin{aligned} E(x_t) &= -S^T(x_t(0))TS(x_t(0)) + 2\sum_{i=1}^{n}\int_0^{(x_t(0))_i} b_i(\sigma)s'_i(\sigma)d\sigma \\ &+ \sum_{k=1}^{K}\frac{1}{\alpha_k}\int_{-\tau_k}^{0}[S(x_t(\theta)) - S(x_t(0))]^T T_k^T \times f_k(\theta)T_k[S(x_t(\theta)) - S(x_t(0))]d\theta, \end{aligned} \tag{4.3}$$

where $(\alpha_1, \cdots, \alpha_K)$ is a sequence of positive numbers such that condition (4.2) is satisfied, and where $f_k(\theta) \in C^1([-\tau_k, 0], R^n)$, $k = 1, \cdots, K$ will be specified later. After changing integration variables, (4.3) can be written as

$$\begin{aligned} E(x_t) &= -S^T(x(t))TS(x(t)) + 2\sum_{i=1}^{n}\int_0^{x_i(t)} b_i(\sigma)\cdot s'_i(\sigma)d\sigma \\ &+ \sum_{k=1}^{K}\frac{1}{\alpha_k}\int_{t-\tau_k}^{t}[S(x(w)) - S(x(t))]^T T_k^T \cdot f_k(w-t)T_k[S(x(w)) - S(x(t))]dw. \end{aligned} \tag{4.4}$$

The derivative of $E(x_t)$ with respect to t along any solution of (3.2) can be computed as

$$\frac{dE(x_t)}{dt} = -\sum_{k=1}^{K}\int_{-\tau_k}^{0}\eta_k(x_t,\theta)^T M_k(x_t,\theta)\eta_k(x_t,\theta)d\theta, \tag{4.5}$$

where $\eta_k(x_t, \theta)^T = [H_0^T, H_k^T, \tilde{G}_k^T]^T$ with H_0, H_k and $\tilde{G}_k$ given by

$$H_0 = -B(x(t)) + T_0S(x(t)) + \sum_{k=1}^{K}T_kS(x(t-\tau_k)), \tag{4.6}$$

$$H_k = T_k[S(x(t-\tau_k)) - S(x(t))], \quad k = 1, \cdots, K, \tag{4.7}$$

$$\tilde{G}_k = T_k[S(x(t+\theta)) - S(x(t))], \quad k = 1, \cdots, K, \tag{4.8}$$

and where $f'(\theta) = \frac{df}{d\theta}(\theta)$,

$$M_k(x_t, \theta) = \begin{bmatrix} \frac{2\alpha_k Q}{\tau_k} & -\frac{Q}{\tau_k} & \frac{QT_k^T f_k(\theta)}{\alpha_k} \\ -\frac{Q}{\tau_k} & \frac{f_k(-\tau_k)}{\tau_k \alpha_k} I & 0 \\ \frac{f_k(\theta) T_k Q}{\alpha_k} & 0 & \frac{f_k'(\theta) I}{\alpha_k} \end{bmatrix}, \tag{4.9}$$

$$Q = A(x(t))S'(x(t)) = S'(x(t))A(x(t)), \tag{4.10}$$

and I denotes the $n \times n$ identity matrix.

We will now show that if the hypotheses of Theorem 1 are satisfied, then $M_k(x_t, \theta)$ is positive definite for all $\theta \in [-\tau_k, 0]$ and all x_t which satisfy equation (3.2), for $k = 1, \cdots, K$. In doing so, we let $U = U_3 U_2 U_1$, where

$$U_1 = \begin{bmatrix} \alpha_k^{-\frac{1}{2}} I & 0 & 0 \\ (2\alpha_k^{\frac{1}{2}})^{-1} I & \alpha_k^{\frac{1}{2}} I & 0 \\ 0 & 0 & \alpha_k^{\frac{1}{2}} I \end{bmatrix}, \quad U_2 = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ -\frac{\tau_k}{2} f_k(\theta) \frac{T_k}{\alpha_k} & 0 & I \end{bmatrix},$$

and

$$U_3 = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & -\frac{1}{2} f_k(\theta) \frac{T_k}{\alpha_k} Q [\frac{f_k(-\tau_k)}{\tau_k} I - \frac{Q}{2\tau_k}]^{-1} & I \end{bmatrix}.$$

It is not difficult to verify that $\tilde{M}_k = U M_k(x_t, \theta) U^T$ is a diagonal matrix. In fact,

$$\tilde{M}_k = diag\{M_{k,1}, M_{k,2}, M_{k,3}\} \tag{4.11}$$

where

$$M_{k,1} = \frac{2Q}{\tau_k}, \tag{4.12}$$

$$M_{k,2} = \frac{f_k(-\tau_k)}{\tau_k} I - \frac{Q}{2\tau_k}, \tag{4.13}$$

and

$$M_{k,3} = f_k'(\theta) I - \frac{f_k(\theta) T_k Q}{2\alpha_k} [(\frac{f_k(-\tau_k)}{\tau_k} I - \frac{Q}{2\tau_k})^{-1} + 2\tau_k Q^{-1}] \frac{QT_k^T f_k(\theta)}{2\alpha_k}. \tag{4.14}$$

It follows that $M_k(x_t, \theta)$ is positive definite if and only if $\tilde{M}_k$ is positive definite, and if and only if $M_{k,1}$, $M_{k,2}$, and $M_{k,3}$ are all positive definite.

If $\tau_k \beta \|T_k\| < \alpha_k$ and, if we chose f_k as

$$f_k(\theta) = l_k \big[\beta \tau_k^2 \frac{\|T_k\|^2}{\alpha_k^2} - \frac{\beta \tau_k \|T_k\|^2 (\tau_k + \theta)}{2\alpha_k^2 - \beta^2 \tau_k^2 \|T_k\|^2}\big]^{-1}, \tag{4.15}$$

where l_k can be any number such that

$$1 > l_k > (2 - \beta^2 \frac{\tau_k^2 \|T_k\|^{2^2}}{\alpha_k})^{-1},$$

then it can be shown that $M_{k,1}$, $M_{k,2}$, and $M_{k,3}$ are all positive definite for all x_t which satisfies (3.2) and for all θ. (Due to space limitations, we omit the details for this derivation.) Furthermore, we know that with the choice of f_k given by (4.15), $M_k(x_t,\theta)$ is positive definite for all $k=1,\cdots,K$, and therefore,

$$\frac{dE(x_t)}{dt} \leq 0 \tag{4.16}$$

along any solution x_t of equation (3.2), where $E(x_t)$ is the energy functional given by (4.3).

Furthermore, applying Invariance Theory (see ch. 4, Lemmas 1.4 and 2.1 of [19], we are able to prove that x_t approaches some equilibrium of system (3.2) as $t \to \infty$. Again, we omit some details due to space limitations. □

V. Local Stability Analysis of Cohen-Grossberg Neural Networks with Multiple Delays

In the preceding section we showed that if $\sum_{k=1}^{K} \tau_k \beta \|T_k\| < 1$, the Cohen-Grossberg neural networks with multiple delays described by equation (3.2)(or equivalently, by equation (1.4)) possess global stability. Since in the implementation of neural networks as associative memories, information is stored in specific asymptotically stable equilibria (called stable memories), good criteria which ensure the asymptotic stability of an equilibrium of (3.2) are of great interest. We address this issue in the present section.

Definition An element $\phi \in C([-\tau,0],R^n)$ is called a *local minimum* of the energy functional E defined by (4.3) if there exists a $\delta > 0$, such that for any $\tilde{\phi} \in C([-\tau,0],R^n)$, $E(\phi) \leq E(\tilde{\phi})$ whenever $|\phi - \tilde{\phi}| < \delta$. □

We are now able to establish the following results.

Theorem 2 Suppose that the conditions of Theorem 1 are satisfied. If x_e is an equilibrium of (3.2), then the following statements are equivalent:

(1) x_e is a stable equilibrium of (3.2);

(2) x_e is an asymptotically stable equilibrium of (3.2);

(3) ϕ_{x_e} is a local minimum of the energy functional E given by (4.3), where $\phi_{x_e} \in C([-\tau,0],R^n)$ such that $\phi_{x_e} \equiv x_e$;

(4) $J(x_e)$ is positive definite, where $J(x)$ is given in equation (J) in Assumption B.

Proof:

a) (1) $\Rightarrow$ (2): Since Assumption B is satisfied, the set of equilibria of system (3.2) is a discrete set by Lemma 3. Therefore, when $\epsilon > 0$ is sufficiently small, there is no other equilibrium in $U(x_e,\epsilon)$, a neighborhood of x_e, given by

$$U(x_e,\epsilon) \triangleq \{x \in R^n : \|x - x_e\| < \epsilon\}. \tag{5.1}$$

Since x_e is a stable equilibrium of (3.2), there exists an $\eta > 0$ such that for any $\phi \in C([-\tau,0],R^n)$ satisfying $|\phi - x_e| < \eta$, $|x_t - x_e| < \epsilon$ for all $t > 0$ where x_t is the solution of

(3.2) with initial condition ϕ. Thus $x_t \in C([-\tau, 0], U(x_e, \epsilon))$ for all t. In view of Theorem 1, x_t will converge to some equilibrium of system (3.2). Since x_e is the only equilibrium of (3.2) in $U(x_e, \epsilon)$, it follows that x_t converges to x_e. Thus, we have shown that x_e is an attractive equilibrium of system (3.2). Therefore, the stable equilibrium x_e of (3.2) is an asymptotically stable equilibrium of system (3.2).

b) (2) ⇒ (3): Since x_e is an asymptotically stable equilibrium of system (3.2), there exists an $\eta > 0$ such that for any $\phi \in C([-\tau, 0], R^n)$ satisfying $|\phi - x_e| < \eta$, x_t converges to x_e where x_t is the solution of (3.2) with initial condition ϕ. Therefore $E(\phi_{x_e}) \leq E(x_t) \leq E(\phi)$ for any $\phi \in C([-\tau, 0], R^n)$ satisfying $|\phi - x_e| < \eta$. Therefore, ϕ_{x_e} is a local minimum of the energy functional E.

c) (3) ⇒ (4): Let $\tilde{E}$ be a function from R^n to R defined by

$$\tilde{E}(x) \triangleq -S^T(x)TS(x) + 2\sum_{i=1}^{n} \int_0^{x_i} b_i(\sigma)s_i'(\sigma)d\sigma. \tag{5.2}$$

Comparing E with $\tilde{E}$, we note that $\tilde{E}$ is a function defined on R^n while E is a functional defined on $C([-\tau, 0], R^n)$. Since ϕ_{x_e} is a local minimum of E, x_e must be a local minimum of $\tilde{E}$. For otherwise, there would exist a sequence $\{x_n\} \subset R^n$ such that $x_n \to x_e$ as $n \to \infty$ and $\tilde{E}(x_n) < \tilde{E}(x_e)$. Let ϕ_{x_n} denote the constant function $\phi_{x_n} \equiv x_n$ in $C([-\tau, 0], R^n)$. Then $|\phi_{x_n} - \phi_{x_e}| \to 0$ as $n \to \infty$, and

$$E(\phi_{x_n}) = \tilde{E}(x_n) < \tilde{E}(x_e) = E(\phi_{x_e}).$$

This contradicts the fact the ϕ_{x_e} is a local minimum of E. Therefore, x_e is a local minimum of $\tilde{E}$. Hence, $\tilde{J}(x_e)$ is positive semidefinite (see, e.g., Theorem 3.6 of [20]), where $\tilde{J}(x)$ is the Hessian matrix of $\tilde{E}$ given by

$$\tilde{J}(x) = \left[\frac{\partial^2 \tilde{E}}{\partial x_i \partial x_j}\right]_{n\times n}. \tag{5.3}$$

It can be shown that

$$\tilde{J}(x) = 2S'(x)J(x)S'(x),$$

where $S'(x) = diag\{s_1'(x_1), \cdots, s_n'(x_n)\}$, and $J(x)$ is given by equation (J) in Assumption B. Therefore, $J(x_e)$ is also positive semidefinite. By Assumption B, $J(x_e)$ is a nonsingular matrix. Thus we have shown that $J(x_e)$ is positive definite.

d) (4) ⇒ (1): We need to prove that x_e is a stable equilibrium of system (3.2). i.e., for any $\epsilon > 0$, there exists a $\delta > 0$ such that for any $\phi \in C([-\tau, 0], R^n)$, if $|\phi - x_e| < \delta$, then $|x_t - x_e| < \epsilon$ where x_t is the solution of (3.2) with initial condition ϕ.

Since $J(x_e)$ is positive definite, then $\tilde{J}(x_e)$ must also be positive definite where $\tilde{J}(x)$ is the Hessian matrix of $\tilde{E}$ given by (5.4). Furthermore,

$$\nabla_x \tilde{E}(x) = 2[-TS(x) + B(x) - b]^T S'(x)$$

where $S'(x)$ is given in part (b). Therefore, $\nabla_x \tilde{E}(x_e) = 0$ since x_e is an equilibrium of (3.2). It follows (by Theorem 3.6 of [20]) that x_e is a local minimum of $\tilde{E}$, i.e., there exists a $\delta_1 > 0$, $\delta_1 < \epsilon$, such that whenever $0 < \|x - x_e\| \le \delta_1$, $\tilde{E}(x_e) < \tilde{E}(x)$. Let $r = \min\{\tilde{E}(x) : \|x - x_e\| = \delta_1\}$. Then it is true that $r > \tilde{E}(x_e)$. Since $E(\phi_{x_e}) = \tilde{E}(x_e)$, it follows that $r > E(\phi_{x_e})$. Note that E is a continuous functional. Therefore, there exists a $\delta \in (0, \delta_1)$ such that whenever $|\phi - x_e| < \delta$, where $\phi \in C([-\tau, 0], R^n)$, we have $E(\phi) < r$. Suppose x_t is any solution of (3.2) with the initial condition ϕ such that $|\phi - x_e| < \delta$. We will show that $|x_t - x_e| < \delta_1 < \epsilon$. For otherwise, there would exist a $t_0 > 0$ such that $\|x_{t_0}(0) - x_e\| = \delta_1$, i.e., $\|x(t_0) - x_e\| = \delta_1$. By the definition of E and $\tilde{E}$, we have $E(x_{t_0}) \ge \tilde{E}(x(t_0)) \ge r$. Therefore, we obtain $E(x_{t_0}) > E(\phi)$ which contradicts the fact that E is monotonically decreasing along any solution of (3.2). Thus, we have shown that x_e is an asymptotically stable equilibrium of system (3.2). □

Remark 4 We note that statement (4) in Theorem 2 is independent of the delays τ_k, $k = 1, \cdots, K$. Therefore, if system (3.2) satisfies Assumptions A and B, and if the condition $\sum_{k=1}^{K} \tau_k \beta \|T_k\| < 1$ is satisfied, then the locations of the (asymptotically) stable equilibria of system (3.2) will not depend on the delays τ_k for $k = 1, \cdots, K$. This is true if in particular $\tau_k = 0, k = 1, \cdots, K$. Therefore, if $\sum_{k=1}^{K} \tau_k \beta \|T_k\| < 1$, then system (3.2) and system (3.1) (obtained by letting $\tau_k = 0$ for $k = 1, \cdots, K$ in (3.2)) will have identical (asymptotically) stable equilibria. We state this in the form of a corollary.

Corollary 1 Under the conditions of Theorem 1, x_e is an (asymptotically) stable equilibrium of system (3.2) if and only if x_e is an (asymptotically) stable equilibrium of system (3.1). This is true if and only if $J(x_e)$ is positive definite, where $J(x)$ is given in equation (J) (in Assumption B). □

Remark 5 Corollary 1 provides an effective criterion for testing the (asymptotic) stability of any equilibrium of Cohen-Grossberg neural networks with multiple delays described by (3.2). This criterion constitutes necessary and sufficient conditions, as long as $\sum_{k=1}^{K} \tau_k \beta \|T_k\| < 1$. □

References

[1] M. Cohen and S. Grossberg, "Absolute stability of global pattern formation and parallel memory storage by competitive neural networks", *IEEE Trans. Syst. Man., Cybern.*, vol. SMC-13, pp. 815-826, 1983.

[2] G. Carpenter, M. Cohen, and S. Grossberg, "Computing with neural networks", *Science*, vol. 37, pp. 54-115, 1987.

[3] B. Kosko, *Neural Networks and Fuzzy Systems*, Prentice Hall, Englewood Cliffs, NJ, 1992.

[4] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", *Proc. Natl. Acad. Sci. U.S.A.*, vol. 79, pp. 2554-2558, 1982.

[5] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons", *Proc. Natl. Acad. Sci. U.S.A.*, vol. 81, pp 3088-3092, 1984.

[6] L. Personnaz, I. Guyon, and G. Dreyfus, "Collective computational properties of neural networks: New learning mechanisms", *Physical Review A*, vol. 34, pp. 4217-4228, 1986.

[7] J. H. Li, A. N. Michel, and W. Porod, "Qualitative analysis and synthesis of a class of neural networks", *IEEE Trans. Circuits Syst.*, vol. CAS-35, no. 8, pp. 976-986, 1988.

[8] L. O. Chua and L. Yang, "Cellular neural networks: Theory", *IEEE Trans. Circuits Syst.*, vol. 35, pp. 1273-1290, 1988.

[9] J. J. Hopfield and D. Tank, "Computing with neural circuits: A model", *Science*, vol. 2, pp. 625-633, 1986.

[10] D. Tank and J. J. Hopfield, "Collective computation in neuronlike circuits", *Scientific American*, 257, 6, pp. 104-114.

[11] J. S. Denker, " Neural networks for computing ", *Amer. Inst. Phys. Conf. 151*, Snowbird, Utah, 1986.

[12] C. M. Marcus and R. M. Westervelt, "Stability of analog neural networks with delay", *Physical Review A*, 39, pp. 347-359, 1989.

[13] B. D. Coleman and G. H. Renninger, "Periodic Solutions of Certain Nonlinear Integral Equations with a Time Lag", , *SIAM J. App. Math.*, vol 31, pp. 111, 1976.

[14] U. an der Heiden, *Analysis of neural networks*, Springer, New York, 1980.

[15] T. A. Burton, "Averaged neural networks", *Neural Networks*, vol. 6, pp. 677-680, 1993.

[16] P. P. Civalleri, M. Gilli, and L. Pandolfi, "On stability of cellular neural networks with delay", *IEEE Trans. Circuits Syst.*, vol. 40, pp. 157-165, 1993.

[17] H. Ye, A. N. Michel, and K. Wang, "Robust stability of nonlinear time-delay systems with applications to neural networks", *Submitted to IEEE Trans. Circuits Syst.*.

[18] H. Ye, A. N. Michel, and K. Wang, "Global stability and local stability of Hopfield neural networks with delays", *accepted by Physical Review E*

[19] J. Hale, *Theory of Functional Differential Equations, Springer-Verlag, 1977* .

[20] C. Goffman, *Calculus of Several Variables*, New York, Harper & Row, 1965.

Combinations of Weak Classifiers: A Case Analysis

Chuanyi Ji
Department of Electrical, Computer and System Engineering
Rensselaer Polytechnic Institute, Troy, NY 12180

Abstract

In this work, classification capability of combinations of weak classifiers is investigated for a linear classification problem. The weak classifiers are weak perceptrons which can do a little better than making random guesses on a training set. We show that when weak classifiers are chosen in an optimal way, the classification error P_g satisfies the following relation $P_g \approx Q(-\frac{2ln\frac{1}{\epsilon}}{\sqrt{\frac{d}{N}+2ln\frac{1}{\epsilon}}}) + O(\frac{1}{\sqrt{2L+1}})$, where $Q(z)$ is the probability distribution function of a standard Gaussian random variable. ϵ is the Bayes error. d is the input dimension. N is the total number of training samples. $2L+1$ is the number of weak classifiers used in a combination. When weak classifiers are chosen in a sub-optimal way so that the mean of weight vectors of weak classifiers deviates from that of Bayes classifier by an angle θ ($0 \leq \theta < \frac{\pi}{2}$), the generalization error satisfies $P_g \approx Q(-\frac{2cos\theta ln\frac{1}{\epsilon}}{\sqrt{\frac{d}{N}+2ln\frac{1}{\epsilon}}}) + O(\frac{1}{\sqrt{2L+1}})$.

1 Introduction

To improve classification performance of a single model (a classifier or a regressor), methods are proposed to combine different models. Specifically, in machine learning, combinations of experts have shown to be able to work at least as well as the best expert in the pool of experts on predicting binary strings[2]. Combinations of neural networks [6][10] and statistical regressors

[1] have also been proposed for both classification and function approximation problems.

Efforts have been made to analyze when and why a combined classifier often has a superior performance compared to a best-selected classifier [6][4] [5]. However, independence assumptions have been used in one way or another in most of the exisiting analysis, which results in unresolved discrepancies between the theory and experimental results[3]. In particular, the questions remained open include

(1) how well can combinations of classifiers possibly do? That is, what is the asymptotic error rate of a combined classifier (when the number of classifiers used in a combination is sufficiently large)?

(2) How does the error rate of a combined classifier approach the asymptotic error rate when more and more classifiers are used in a combination? Is the error rate reduced monotonically when more and more classifiers are combined? How fast is the convergence?

(3) What are the advantages of combinations of classifiers over conventional methods?

The goal of this work is to formally address these problems in a context of combinations of weak classifiers, where weak classifiers are the classifiers which can do a little better than random guessing. Combinations of a large number of weak classifiers were initially proposed by Kleinberg [4] and Shapire [8] and have been applied to optical character recognitions[9][3]. The motivation for us to consider weak classifiers as opposed to well-trained classifiers in this context is that there exist many such weak classifiers, and they are computationally cheap to obtain as well.

In order to obtain analytical results, we will focus on a simple problem in this work: approximating an optimal linear classifier through combining a large number of weak perceptrons. The insight gained, however, can shed light on more difficult problems.

2 The Problem

The problem we consider is a two-class classification problem, where each class conditional density of samples is spherically symmetric in d dimensions about one of the two vertices μ_i $(i = 1, 2)$ of a binary hypercube. Without loss of generality, we assume ${\mu_1}^T = [1, ..., 1]$ and $\mu_2 = -\mu_1$. By flipping each bit of μ_1 independently with the probability p $(0 < p \leq \frac{1}{2})$, a new vector $x_1^{(1)} \in R^d$ is generated and assigned to Class 1 with a label $+1$. The notation $x_j^{(m)}$ represents the mth sample $(m \geq 1)$ belonging to Class j $(j = 1, 2)$. By repeating the process, a set of independent binary vectors belonging to Class 1, $\{x_1^{(m)}\}$, can be generated. These vectors will form a cluster around

μ_1. Similarly, if bits of μ_2 are filpped in the same way, a set of independent samples $\{x_2^{(m)}\}$ $(m \geq 1)$ belonging to Class 2 can be generated, which form another cluster around μ_2. Assume that the prior probabilities of the two classes are equal, i.e., $\Pr(\Omega_1) = \Pr(\Omega_2)$, then the two clusters of samples from two classes are symmetrical.

Mathematically, a sample $x_j^{(m)}$ can be considered as a random vector, the i-th element $x_{ij}^{(m)}$ of which can be expressed as

$$x_{ij}^{(m)} = \begin{cases} (-1)^j & \text{with probability } p, \\ -(-1)^j & \text{with probability } (1-p), \end{cases}$$

for $0 < p \leq \frac{1}{2}$, $i \in [1, d]$, $m \geq 1$ and $j = 1, 2$.

It is easy to see that the optimal classifier for this problem is linear which is essentially $\mu_1 - \mu_2$. The Bayes error is greater than zero, and small if p is close to $\frac{1}{2}$.

3 Weak Classifiers

A training set consisting of N iid (labeled) training samples is used to obtained weak classifiers. Roughly speaking, weak classifiers need to satisfy two conditions. First, an average classification rate of each weak classifier on the training set is $\frac{1}{2} + \frac{1}{\nu}$, where $\nu >> 2$ is a so-called weakness factor which measures the strength of individual weak classifiers. Second, a collection of weak classifiers should be powerful enough to learn an entire training set. More specifically, a combined classifier should be able to correctly classify any training sample in a training set with a high probability.

To be able to quantify the performance of combinations of weak classifiers later, we define an optimality measure of weak classifiers through a quantity θ. Let μ be the expected value of a d-dimensional weight vector of a weak classifier (assuming weight vectors of all weak classifiers have the same expected value), and w_b be the d-dimensional weight vector for the Baysian classifier. Then θ is an angle between μ and w_b, i.e.,

$$cos\theta = \frac{\mu^T w_b}{|| \mu |||| w_b ||}. \tag{1}$$

If $\theta = 0$, the weak classifiers are optimal; otherwise, they are sub-optimal.

A combined classifier is obtained through a simple majority vote of the outputs of weak classifiers.

4 Results

Analytical results have been derived based on generalization error [1] of combined classifiers for both optimal and sub-optimal weak classifiers. The first case provides the optimal performance of the combined classifier, and the second one shows how the performance of a combined classifier deviates from the optimum in general.

4.1 Optimal Weak Classifiers

For this case, the weak classifiers are constructed through learning a small fraction of a training set at a time based on sampling a training set with replacement. More specifically, $2M_1$ ($M_1 \ll N$) training samples are drawn randomly from a training set each time. These $2M_1$ samples are called "cares", since they will be used to obtain weights of a weak classifier, whereas the rest of the samples in the training set are called "don't-cares". Then weights of a weak classifier are obtained as the difference between two sample means for the samples in two distinct classes using only these $2M_1$ samples. That is, let $x_1^{(1)}, ..., x_1^{(M_1)}$ be M_1 d-dimensional samples in Class 1, and $x_2^{(1)}, ..., x_2^{(M_1)}$ be M_1 samples in Class 2. The weight vector w_l of the l-th weak classifier can be represented as

$$w_l = \frac{1}{2M_1} \sum_{m=1}^{M_1} (x_1^{(m)} - x_2^{(m)}), \tag{2}$$

where $1 \leq l \leq 2L + 1$.

The $2M_1$ samples are then put back into the training set for reuse.

It is easy to verify that the weights of weak classifiers constructed this way are indeed optimal, since the Bayes classifier for classifying samples in two symmetrical clusters is simply the difference between two means which are the two centers of the clusters. On the other hand, since $2M_1 \ll N$, the majority of the training samples are left to be don't-cares. Therefore, the training error rate of each weak classifier is still high on the entire training set. It can be shown that when $2M_1$ is within the statistical capacity of a perceptron, the weakness factor $\frac{1}{\nu} = O(\sqrt{\frac{M_1}{d}})$, which is small for $M_1 \ll d$.

To obtain the error rate for the combined classifier, efforts are made to deal with dependent outputs of weak classifiers. It can be shown that when the number of weak classifiers $2L + 1$ and the input dimension d are large, but $2L + 1 \ll d$, properties of exchangable joint Gaussian random variables can be used to obtain the result given in the theorem below.

---

[1] the probability of incorrect classification of a randomly-drawn sample which is not included in a training set

Theorem 1 *Let ϵ be the Bayes error. Assume that the input dimension d, the number of weak classifiers used in a combination $2L+1$, the number of cares $2M_1$ and the total number of training samples N are large, but $2M_1 \ll d \ll N$ and $2L+1 \ll d$. Then the error rate (the generalization error), denoted as P_g, of a combined classifier satisfies the relationship*

$$P_g \approx Q\left(-\frac{2ln\frac{1}{\epsilon}}{\sqrt{\frac{d}{N}+2ln\frac{1}{\epsilon}}}\right)+O\left(\frac{1}{ln\frac{1}{\epsilon}\sqrt{(2L+1)}}\sqrt{\frac{d}{2M_1}}\right), \tag{3}$$

where $Q(z)=\int_{-\infty}^{z}\frac{1}{\sqrt{2\pi}}e^{-\frac{t^2}{2}}dt$. $\Omega()$ and $O()$ represent quantities in the order of () used for lower and upper bounds, respectively. This relationship can also be represented as

$$P_g \approx \epsilon + O\left(\frac{d}{N}\right)+O\left(\frac{1}{\sqrt{2L+1}ln\frac{1}{\epsilon}}\sqrt{\frac{d}{2M_1}}\right). \tag{4}$$

4.2 General Weak Classifiers

For more general cases, weights of weak classifiers w_l's are assumed to be independent and randomly drawn from a distribution, where the means and the variances of the elements of w_l can be represented by two vectors: $\mu^T = (\mu_1, ..., \mu_d)$ and $\sigma^T = (\sigma_1, ..., \sigma_d)$. Then similar analysis can be carried over to obtain the generalization error, P'_g, of the combined classifier as described below.

Theorem 2 *Assume $cos\theta \geq 0$, where $cos\theta$ is defined through Equation (1). Then the generalization error of the combined classifier can be represented as*

$$P'_g \approx Q\left(-\frac{2cos\theta ln\frac{1}{\epsilon}}{\sqrt{\frac{d}{N}+2ln\frac{1}{\epsilon}}}\right)+O\left(\frac{1}{\sqrt{\rho(2L+1)}}\sqrt{\frac{d}{2M_1}}\right), \tag{5}$$

where $\rho \approx \frac{||\mu||^2}{||\mu||^2+||\sigma||^2}$ for d large.

It is clear that the optimal choice of weak classifiers are simply a special case for $\theta = 0$ and $\rho \approx 2ln\frac{1}{\epsilon}\frac{2M_1}{d}$.

5 Discussions

The above results show that the optimal asymptotic error rate of a combined classifier is the Bayes error ϵ plus an additive term $O(\frac{d}{N})$. This error rate represents the optimal performance of a combined classifier on a finite training set of size N when the number of weak classifiers approaches infinity. In general, the asymptotic error is larger than the optimal error rate depending on the amount of deviation of the average weak classifier from the Bayes classifier. In some sense, the asymptotic error represents the bias in the combinations of weak classifiers, where the bias can only be circumvented through choosing proper methods to construct weak classifiers.

When the number of weak classifiers used in a combination is finite, an additional term is added to the asymptotic error. This terms shows that the error rate of a combined classifier with $2L + 1$ weak classifiers approaches its asymptotic error monotonically at a rate $\Omega(\frac{1}{\sqrt{2L+1}})$. This additional term is like the variance in combinations of weak classifiers. It indicates that the uncertainty introduced by using a finite number of (random) weak classifiers can be overcome by adding more and more weak classifiers. In addition, the term $O(\frac{1}{\sqrt{2L+1} ln \frac{1}{\epsilon}} \sqrt{\frac{d}{2M_1}})$ for optimal weak classifiers shows that the weaker the weak classifiers (the smaller the $\frac{2M_1}{d}$), the more weak classifiers are needed to achieve the same error rate. In the meantime, the smaller the Bayes error ϵ, the better the separation among samples in two distinct classes, and the fewer weak classifiers are needed to achieve the same error rate.

To identify advantages of combinations of weak classifiers, we choose k-nearest neighbor classifiers for comparison. It has been shown in [7] that k-nearest neighbor classifiers suffers the curse of dimensionality. That is, the error rate of an k-nearest neighbor classifier will approach its asymptotic error at an exponentially slow rate when the input dimension increases. This will be true even for a simple linear classification problem similar to the one we consider in this work. For combinations of weak classifiers, however, the error rate approaches the asymptotic error polynomially in terms of input dimension, the number of training samples as well as the number of weak classifiers. This is due to the architecture of weak perceptrons which is non-local. However, the weakness of those perceptrons introduces locality to the weak classifiers.

It should be noted at last that the above analysis indicates that outputs of weak classifiers as well the errors made are in general statistically dependent. It is the dependence among outputs of weak classifiers that may cause both the possible performance degradation in terms of an asymptotic error, and a convergence rate no faster than $O(\frac{1}{\sqrt{2L+1}})$ for this problem when more and more weak classifiers are used.

Acknowledgement

Support from National Science Foundation (ECS-9312504 and (CAREER) IRI-9502518) is gratefully acknowledged.

References

[1] L. Breiman, "Stacked Generalization," *Technical Report, TR-367, Department of Statistics, University of California, Berkeley*, Aug., 1992.

[2] N. Cesa-Bianchi, Yoav Freud, D.P. Helmbold, D. Haussler R.E. Shapire and M.K. Warmuth, " How to Use Expert Advice," *Proceeding of Foundation of Computer Science*, 382-391, 1993.

[3] E.M. Kleinberg and T. Ho, "Pattern Recognition by Stochastic Modeling," *Proceeding of Third International Workshop on Frontiers in Handwriting Recognition*, 175-183, Buffalo, May 1993.

[4] E.M. Kleinberg, "Stochastic Discrimination," *Annel of Mathematics and Artificial Intelligence*, vol.1, 207-239, 1990.

[5] R. Meir, preprint.

[6] M.P. Perrone and L.N. Cooper, "When Networks Disagree: Ensemble Method for Neural Networks," Chap 10, *Artificial Neural Networks for Speech and Vision*, 1993.

[7] D. Psaltis, R.R. Snapp and S.S Venkatesh, "On The Finite Sample Performance of The Nearest-Neighbor Classifier," *IEEE Trans. Information Theory*, v40 (3), 820-837, 1994.

[8] R. Shapire, "The Strength of Weak Learnability," *Machine Learning*, Vol.5, No.2, 197-227, 1990.

[9] Harris Drucker, R. Shapire and Patrice Simard, "Improving Performance in Neural Networks Using Boosting Algorithm," *Proceeding Neural Information Processing Sympothum*, 1993.

[10] D. Wolpert, "Staked Generalization," *Neural Networks*, vol. 5(2), 241-259, 1992.

FANOUT SPLITTING IN NONBLOCKING COPY NETWORKS WITH SHARED BUFFERING

A. Chandra[1], H. Saidi[1], P.S. Min[1], and M.V. Hegde[2]
1: Washington University in St. Louis
2: Louisiana State University

1 Introduction

An important issue in ensuring high performance in the multicast switches [1] is the development of a copying policy that enables the replication of packets with high efficiency. The main purpose of this paper is to investigate the effects of a certain copying policy which can be used to improve the performance such as packet loss probability, delay, and throughput. This policy, which we will term as *fanout splitting*, allows the fanout request of a single packet to be satisfied over multiple time slots.

We concentrate in this paper on copy networks exhibiting the internally nonblocking property and utilizing shared buffering.

A copy network can be viewed abstractly as a functional block with N input channels and N output channels. For convenience, input and output channels are each labeled between 0 and $N-1$, starting from the top. At each input channel, a fixed length packet may appear with probability p per time slot, independently of events in any other time slot, and independently of events in any other input channel.

An arriving packet has associated with it the fanout request value, F, which corresponds to the number of copies to be made by the copy network. The fanout request values for different packets are assumed to be independent and identically distributed (i.i.d.) random variables with distribution $\mathcal{F}$ according to which k copies are requested with probability f_k ($\sum_{i=1}^{N} f_k = 1$).

Packets are replicated in the copy network and each output channel allows at most one of the replicated copies to depart from the copy network in a time slot. Thus no more than N copies of the packets which are present can leave in the same time slot. In the assumed paradigm, the specification of output channels for the replicated packets to exit from is not needed.

2 Shared Buffering with Fanout Splitting

Shared buffering is achieved by providing a common buffer module of size b that is accessible from all input channels. In each time slot, packets in the buffer module are considered first for copying (in a FIFO manner) and then the new packets arriving at the input channels are considered (according to an order $e_0, e_1, \ldots, e_{N-1}$, where $e_0, e_1, \ldots, e_{N-1}$ is some permutation of $\{0, 1, 2, \ldots, N-1\}$). In order to ensure that no input channel is given any priority, we assume that the order $e_0, e_1, \ldots, e_{N-1}$ is fair in the sense that $Pr(e_l = j) = \frac{1}{N}$ for all $0 \leq l, j \leq N-1$. Thus, packets arriving at the input may enter the buffer module in a different order from one time slot to the next, in such a manner as to treat all input channels fairly.

Newly arrived packets in a time slot are eligible for copying in the same time slot, and are lost at the end of the time slot if buffers cannot be made available for them after the copying process is completed.

Fanout splitting is a copying policy wherein the fanout request value of a packet is allowed to be fulfilled over two consecutive time slots. Suppose that after copying a certain number of packets during a time slot, there are only enough uncommitted output channels to satisfy part of the fanout request of a packet. Instead of delaying the replication of this packet in its entirety, fanout splitting allows as many copies as can be accommodated during this time slot for the packet, and the remaining number of requested copies to be replicated in the next time slot.

We utilize the random variable R_t which is defined to be the fanout request value to be satisfied in the beginning of time slot t, i.e., the fanout request value of the head-of-line packet in the buffer as a result of the copying process in time slot $t-1$. As an illustration, assume that during time slot $t-1$, after a certain number of packets are copied, there remain n output channels. Suppose that the next packet to be copied has a fanout request value of $\tilde{F}$ where $\tilde{F} > n$. With the fanout splitting scheme, n replicates of this packet can exit from the remaining n output channels in time slot $t-1$. The remaining $\tilde{F} - n$ replicates need to be transmitted in the beginning of time slot t, which implies $R_t = \tilde{F} - n$.

If all available packets are copied in time slot $t-1$ (and thus no packet remains in the buffer), we set $R_t = 0$. We note that R_t is associated with the packet that is at the head position of the buffer prior to new arrivals in time slot t. The fanout request value of a newly arriving packet, which happens to arrive to an empty buffer module and thus assumes the head position, is not implied.

Figure 1 shows an example of the fanout splitting scheme; the packets labeled as A, B, and C with the fanout request values 2, 4, and 1 respectively, are copied in their entirety in time slot t, whereas the packet labeled as D with the fanout request value 3 is partially copied generating 1 replicate in time slot t and the 2 remaining replicates are made in time slot $t+1$.

3 Distributions of Buffer Occupancy and Head-of-Line Fanout Request

We now develop an analytical model for the queueing behavior of the buffer module. In the beginning of time slot t, let R_t be the head-of-line fanout request value (i.e., the value which is determined at the conclusion of the copying process in time slot $t-1$), and let V_t be the number of packets in the buffer module. Under the assumed notation, the condition $(R_t = 0)$ is equivalent to the condition $(V_t = 0)$. (No packet is in the buffer module, so no head-of-line fanout request value.) Also, $(R_t > 0)$ is equivalent to $(V_t > 0)$. (The head-of-line packet with the fanout request value of R_t indicates that there is at least one packet in the buffer module.) Let U_t be the number of newly arrived packets from the N input channels. Thus, $P(U_t = u) = \binom{N}{u} p^u (1-p)^{N-u}$ and U_t is independent of V_t.

Now, given (R_t, V_t), the values of (R_{t+1}, V_{t+1}) depend on U_t, and the fanout request values of the packets which are distributed according to $\mathcal{F}$ independently of (R_t, V_t). Hence the (R_t, V_t) form a Markov chain, which we call M_A. Since the distribution $\mathcal{F}$ and the probability p do not depend on time, M_A is time homogeneous. We note first that the transition probabilities of M_A can be written as

$$\begin{aligned} &P(R_{t+1} = r', V_{t+1} = v' | R_t = r, V_t = v) \\ &= \sum_{u=0}^{N} P(R_{t+1} = r', V_{t+1} = v' | R_t = r, V_t = v, U_t = u) P(U_t = u). \end{aligned} \tag{1}$$

To proceed with determining $P(R_{t+1} = r', V_{t+1} = v' | R_t = r, V_t = v, U_t = u)$, we employ another Markov chain, M_B, with state space $\{N, N-1, \dots, 1, 0, -1, -2, \dots, -N\}$, whose states reflect the level of utilization in output channels in a time slot. A description of M_B, whose states will be referred to as *output states*, is presented next.

In the beginning of time slot t, all N output channels are available for some replicated packets to exit from and the state of M_B is N. If $R_t = r$, the first packet to be copied utilizes r output channels, and therefore, there are $N - r$ output channels available for the replicates of the subsequent packets. We say that M_B is in output state $N - r$. Let the fanout request value of the next packet to be copied (if there is one) be k. If $k \leq N - r$, we can make copies for this packet, thereby committing k additional output channels and the output state (i.e., the state of M_B) becomes $n = N - r - k$, implying that there are $N - r - k$ uncommitted output channels.

Assume that M_B is in state n. If the next packet has a fanout request value of k' such that $n \geq k'$, the output state then goes to $n - k'$. On the other hand, if $n < k'$, then only n copies of this packet can be replicated in this time slot and the output state goes to $-(k' - n)$ implying that the first fanout request value to be satisfied in time slot $t + 1$ (i.e. R_{t+1}) is $k' - n$. When this happens, no additional packet is considered for copying in time slot t.

Given the independent and identical nature of the fanout request values at N input channels, M_B can be seen to be a homogeneous Markov chain, and the stochastic matrix, P, for M_B can be written as,

$$P = \begin{pmatrix} P_{N,N} & P_{N,N-1} & P_{N,N-2} & P_{N,N-3} & \cdots & P_{N,-N} \\ P_{N-1,N} & P_{N-1,N-1} & P_{N-1,N-2} & P_{N-1,N-3} & \cdots & P_{N-1,-N} \\ P_{N-2,N} & P_{N-2,N-1} & P_{N-2,N-2} & P_{N-2,N-3} & \cdots & P_{N-2,-N} \\ \cdot & \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdots & \cdot \\ P_{N-n,N} & P_{N-n,N-1} & P_{N-n,N-2} & P_{N-n,N-3} & \cdots & P_{N-n,-N} \\ \cdot & \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdots & \cdot \\ P_{0,N} & P_{0,N-1} & P_{0,N-2} & P_{0,N-3} & \cdots & P_{0,-N} \\ \cdot & \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdots & \cdot \\ P_{-N,N} & P_{-N,N-1} & P_{-N,N-2} & P_{-N,N-3} & \cdots & P_{-N,-N} \end{pmatrix} \tag{2}$$

where $P_{n,m}$ is the transition probability of the output state from n to m. For example, if

$N = 4$,

$$P = \begin{pmatrix} 0 & f_1 & f_2 & f_3 & f_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & f_1 & f_2 & f_3 & f_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & f_1 & f_2 & f_3 & f_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & f_1 & f_2 & f_3 & f_4 & 0 \\ 0 & 0 & 0 & 0 & 0 & f_1 & f_2 & f_3 & f_4 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

The matrix P, which is associated with M_B, can be used to calculate the transition probabilities of the Markov chain M_A, which describe the evolution of (R_t, V_t). We first decompose P into P^1 and P^2 such that $P = P^1 + P^2$, when the fanout request of a packet is completely satisfied, a transition is made in the output state according to P^1, and for each of the other packets (i.e., the ones to remain in the buffer module until time slot $t+1$ or to be lost due to buffer overflow), an output state transition occurs according to P^2. For $N = 4$,

$$P^1 = \begin{pmatrix} 0 & f_1 & f_2 & f_3 & f_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & f_1 & f_2 & f_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & f_1 & f_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & f_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$P^2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & f_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & f_3 & f_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & f_2 & f_3 & f_4 & 0 \\ 0 & 0 & 0 & 0 & 0 & f_1 & f_2 & f_3 & f_4 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Let $\psi_{nm}(x, y|z)$ be the probability of going from output state n to output state m in z transitions with x transitions according to P^1 and y transitions according to P^2. (Thus $z = x + y$.) Then, with $n, m \in \{N, N-1, N-2, \ldots, 1, 0, -1, -2, \ldots -N\}$, the following equations describe recursive relationships amongst the ψ's:

$$\begin{aligned} \psi_{nm}(0, 0|0) &= \delta_{nm}, \qquad (3) \\ \psi_{nm}(0, y|z) &= \sum_{s=-N}^{N} \psi_{ns}(0, y-1|z-1) P^2_{s,m}, \end{aligned}$$

$$\begin{aligned}\psi_{nm}(x,0|z) &= \sum_{s=-N}^{N} \psi_{ns}(x-1,0|z-1)P^1_{s,m},\\ \psi_{nm}(x,y|z) &= \sum_{s=-N}^{N} \psi_{ns}(x-1,y|z-1)P^1_{s,m} + \sum_{s=-N}^{N} \psi_{ns}(x,y-1|z-1)P^2_{s,m}\end{aligned}$$

where δ_{nm} is 1 if $n = m$, and is 0 otherwise. $P^1_{s,m}$ and $P^2_{s,m}$ are the $(N-s)^{th}$ row, $(N-m)^{th}$ column elements of P^1 and P^2 respectively. Then, noting that $(r' = 0)$ is equivalent to $(v' = 0)$, and $(r = 0)$ is equivalent to $(v = 0)$, we have

$$\begin{aligned}&P(R_{t+1} = r', V_{t+1} = v' | R_t = r, V_t = v, U_t = u)\\ &= \begin{cases} \sum_{l=0}^{N} \psi_{N,l}(u,0|u), & v'=0, v=0;\\ \psi_{N,-r'}(u-v',v'|u), & 0<v'<b, v=0;\\ \sum_{k=b}^{u} \psi_{N,-r'}(u-k,k|u), & v'=b, v=0;\\ \sum_{l=0}^{N} \psi_{N-r,l}(u+v-1,0|u+v-1), & v'=0, v>0;\\ \psi_{N-r,-r'}(u+v-v'-1,v'|u+v-1), & 0<v'<b, v>0;\\ \sum_{k=b}^{u+v-1} \psi_{N-r,-r'}(u+v-k-1,k|u+v-1), & v'=b, v>0. \end{cases}\end{aligned} \tag{4}$$

The transition probabilities of M_A can be determined as

$$\begin{aligned}&P(R_{t+1} = r', V_{t+1} = v' | R_t = r, V_t = v)\\ &= \sum_{u=0}^{N} P(R_{t+1} = r', V_{t+1} = v' | R_t = r, V_t = v, U_t = u) \binom{N}{u} p^u (1-p)^{N-u},\end{aligned} \tag{5}$$

where $P(R_{t+1} = r', V_{t+1} = v' | R_t = r, V_t = v, U_t = u)$ can be determined from (4).

With the transition probabilities given by (5), M_A can be seen to be aperiodic and irreducible, and from the finite cardinality of the buffer module, we see that there exists a unique stationary distribution for M_A.

4 Packet Loss Probability, Throughput and Delay

We now calculate the probability of packet loss due to buffer overflow in the following manner. Assume that, in time slot t, a tagged packet enters the copy network at a certain input channel and after traversing through the (fictitious) randomizing network, it appears at e_l. There are four possible dispositions for the tagged packet during the time slot: it can be fully copied and all its copies exit the copy network, it can be partially copied and remains at the head of the buffer module with the fanout request value R_{t+1}, it can be buffered at a position other than the head of the buffer module, or it can be lost due to buffer overflow.

Define $m_{lv'r'}$ to be the probability that the packet arriving at e_l moves to the v'^{th} position at the end of time slot t and that R_{t+1} becomes r'. In this notation, packets which are copied and transmitted in time slot t may be thought of as moving to the 0^{th} position in the buffer. Denote by $U_{t,l}$, the random variable describing the number of new packets at input channels $e_0, e_1, \ldots, e_{l-1}$ in time slot t. By conditioning on the event that the state of M_A in the beginning of time slot t is $R_t = r$ and $V_t = v$, and the value of $U_{t,l}$ is β, we can write

$$\begin{aligned} m_{lv'r'} &= \sum_{r,v,\beta} P(R_{t+1} = r', V_{t+1} = v' | R_t = r, V_t = v, U_{t,l} = \beta) P(R_t = r, V_t = v) P(U_{t,l} = \beta) \\ &= \sum_{r,v,\beta} P(R_{t+1} = r', V_{t+1} = v' | R_t = r, V_t = v, U_t = \beta + 1) P(R_t = r, V_t = v) P(U_{t,l} = \beta) \end{aligned}$$

where $P(R_t = r, V_t = v)$ is known from the analysis of M_A and $P(U_{t,l} = \beta)$ is determined from the $B(l, p)$ distribution of the binomial random variable, $U_{t,l}$.

We denote the probability that this packet is transmitted as m_{lT}, the probability that this packet is partially copied and buffered at the head of the buffer module as m_{lP}, and the probability that this packet is lost due to buffer overflow as m_{lL}. Then, $m_{lT} = m_{l00}$, $m_{lP} = \sum_{r'} m_{l1r'}$, and $m_{lL} = \sum_{k>b} \sum_{r'} m_{lkr'}$. Note that we now have extended the scope of $P(R_{t+1} = r', V_{t+1} = v' | R_t = r, V_t = v, U_t = \beta + 1)$ to include the case when $v' > b$ (in the equation for m_{lL}), which is not shown in (4). For $v' > b$,

$$\begin{aligned} & P(R_{t+1} = r', V_{t+1} = v' | R_t = r, V_t = v, U_t = u) \\ &= \begin{cases} \psi_{N,-r'}(u - v', v' | u), & v = 0 \\ \psi_{N-r,-r'}(u + v - v' - 1, v' | u + v - 1), & v > 0. \end{cases} \end{aligned} \tag{6}$$

Since the probability of an input channel being e_l is equal for $l = 0, 1, 2, \ldots, N-1$, the average packet loss probability, L, and the average throughput, T, become,

$$L = \frac{1}{N} \sum_{l=0}^{N-1} m_{lL} \tag{7}$$

$$T = p(1 - L)E(F). \tag{8}$$

To determine the average delay, D, consider again the tagged packet which appeared at e_l. Define g_l to be the number of time slots until the beginning of transmission for the tagged packet. Then,

$$\begin{aligned} g_l &= 0 \text{ with probability } m_{lT} \\ g_l &= 1 \text{ with probability } m_{lP} \\ &= 1 + g_{vr} \text{ with probability } m_{lvr}, \quad v = 2, 3, \ldots, \quad b, r = 1, 2, \ldots, N, \end{aligned} \tag{9}$$

where g_{vr} is the number of time slots until transmission for the packet at the v^{th} position in the buffer module with $R_t = r$. Noting that g_{1r} equals zero for all r, $E(g_l)$ can be written as $E(g_l) = \sum_{r,v>0} (1 + E(g_{vr})) m_{lvr}$, where $E(g_{vr})$ is determined by,

$$E(g_{vr}) = \sum_{r',v'>0} P(R_{t+1} = r', V_{t+1} = v' | R_t = r, V_t = v, U_t = 0) \cdot (1 + E(g_{v'r'})). \tag{10}$$

Equation (10) needs to be solved for $E(g_{vr})$, and using these values, $E(g_l)$ can be evaluated. Then, the delay D becomes,

$$D = \frac{\sum_{l=0}^{N-1} E(g_l)(1 - m_{lL})}{\sum_{l=0}^{N-1}(1 - m_{lL})}. \tag{11}$$

5 Numerical Results and Conclusions

We consider all variables at steady state, and we omit the subscript t from R_t and V_t. Before we establish the validity of the analyses given in Sections 3 and 4, it is interesting to note the distribution, $\mathcal{R}$, of R, as compared to the distribution, $\mathcal{F}$, of F. Figure 2 shows $\mathcal{R}$ recorded from the simulated data under different $\mathcal{F}$, wherein for the purpose of comparing to $\mathcal{F}$, $\mathcal{R}$ is normalized by $(1 - P(R = 0))$. Also shown in Figure 2 are the shapes of $\mathcal{R}$ if fanout splitting is not allowed. We observed during our numerical study that the distinguishing shapes in $\mathcal{R}$ in Figure 2 have a significant impact on packet loss probability, delay, and throughput. Without explicit consideration of the differences between $\mathcal{R}$ and $\mathcal{F}$ (i.e., assuming that R and F are identically distributed), the numerical results deviate significantly from the simulated values.

Figure 3 compares the results of the analyses with the simulated data with $\mathcal{F}$ set to be a truncated geometric distribution where

$$f_k = \frac{(1 - \lambda)^{k-1}\lambda}{\sum_{h=1}^{N}(1 - \lambda)^{h-1}\lambda}.$$

λ is a parameter such that $0 \le \lambda \le 1$ and is set to result in an appropriate $E(F)$ value.

We see from the figure that our analyses are accurate for all values of ρ $(= p \cdot E(F))$. The accuracy of the throughput follows from the accuracy of the packet loss probability and the relation (8). In fact, in all the numerical studies we conducted, we observed similar accuracy in our results.

In Figure 4, we observe the effects of fanout splitting on the performance of copy networks with the truncated geometric fanout request values. As suspected, the performance improves significantly with fanout splitting, especially where the packet loss probability is less than 10^{-6}.

Figure 5 illustrates the performance of copy network at the channel level; in the figure, the packet loss probabilities are shown for e_l with differing l. Figure 5 confirms the needs for randomizing the service order among the input channels. Without such randomization, there is significant disparity in the performance among the input channels. For example, with truncated geometric fanout request values, it can be seen from Figure 5 that for all values of ρ, there is more than three orders of magnitude difference in the packet loss probability between channel 1 and channel 15.

It is also interesting to observe the effect of variance in the fanout request values on the performance of the copy network. Let F_v be the random variable corresponding for the fanout request value such that for $v = 1, 2, \ldots, \lfloor \frac{N}{2} \rfloor$

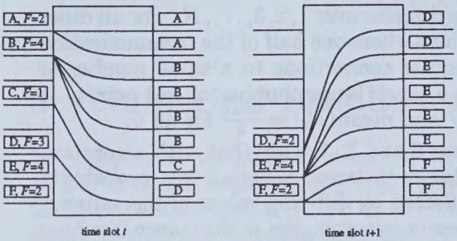
$$\begin{aligned} P(F_v = k) &= \frac{1}{2v} \quad k = 1, 2, \ldots, v \\ &= 0 \quad k = v + 1, v + 2, \ldots, N - v \\ &= \frac{1}{2v} \quad k = N - v + 1, N - v + 2, \ldots, N. \end{aligned} \tag{12}$$

For even N with $v = \frac{N}{2}$, (12) is a uniform distribution over $1, 2, 3, \ldots, N$. For all other values of N and v, F_v represents bimodal situations where one half of the communication sessions established in the multicast switch require connections to a small number of end-points and the other half are connected to a much larger number of end-points. It is easy to see that $var(F_v) > var(F_{v'})$ for $v < v'$ and $mean(F_v) = \frac{N+1}{2}$ for all v.

The effect of different F_v's are seen in Figures 6 and 7. We see that with a constant average value, the performance of the copy network in terms of packet loss probability (and, therefore, throughput) and delay is not affected by differing values of the variance in any significant way. Thus, at least for the present example, the performance is robust with respect to different variance values. Such a property is desired since the performance of the copy network can be monitored by measurements of channel utilization only.

References

[1] Min, P.S., Saidi, H., and Hegde, M.V., "A Nonblocking Architecture For Broadband Multi-Channel Switching," *IEEE/ACM Transactions on Networking*, Vol. 3, No. 2, pp. 181-198, April 1995.



A, F=2
B, F=4
C, F=1
D, F=3
E, F=4
F, F=2
A
A
B
B
B
B
C
D
time slot t
D
D
B
B
B
B
F
F
D, F=2
E, F=4
F, F=2
time slot t+1

Figure 1. Example of Fanout Splitting

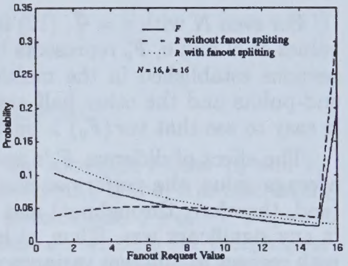


Figure 2. Fanout Request Value (*F*: Truncated Geometric, $E(F) = 8$)

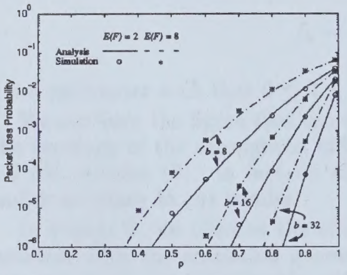


Figure 3a. Packet Loss Probability with $N = 16$

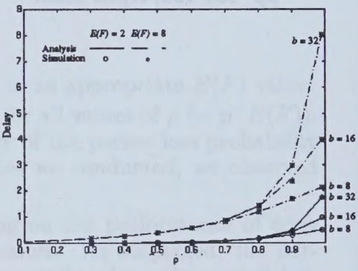


Figure 3b. Delay with $N = 16$

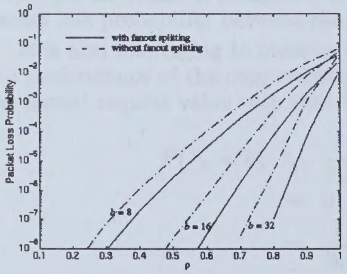


Figure 4a. Packet Loss Probability Comparison with $N = 24$ and $E(F) = 2$

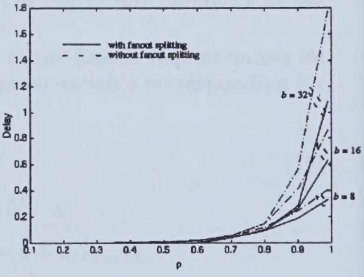


Figure 4b. Delay Comparison with $N = 24$ and $E(F) = 2$

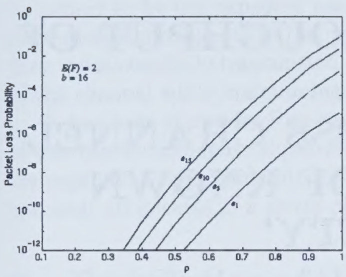


Figure 5. Channel Level Performance with $N = 16$

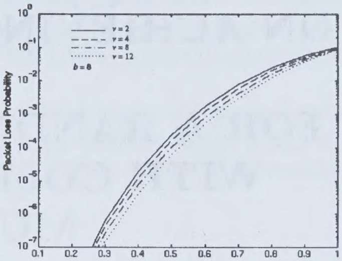


Figure 6. Packet Loss Probability for Bimodal Distribution with $N = 24$ and $E(F) = 12.5$

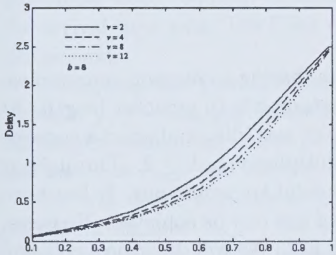


Figure 7. Delay for Bimodal Distribution with $N = 24$ and $E(F) = 12.5$

ON ACHIEVING A THROUGHPUT OF ONE FOR A RANDOM ACCESS CHANNEL WITH COLLISIONS OF KNOWN MULTIPLICITY*

Shivendra S. Panwar
Polytechnic University, Brooklyn, NY
Walter Grote
UTFSM, Valparaiso, Chile

October 2, 1995

Abstract- We consider the problem of a group of users sharing a common communication channel. The channel is slotted and the slot size corresponds to a packet length. At the end of each slot, every user is informed whether the slot was idle, contained a successful packet transmission, or experienced a collision of multiplicity k, $k \geq 2$. Throughput is defined to be the expected fraction of slots with successful transmissions. It has been shown, via a non-constructive proof, that a throughput of one can be achieved. However, the best throughput achieved thus far is approximately 0.532. We will describe a collision resolution algorithm (CRA) which achieves, in the limit, a throughput of one.

Index Terms- Random Access Communications, Collision Resolution Algorithms

1 Introduction

The problem of a group of users sharing a common channel has been widely investigated [8]. Much of the work assumes that, in the event of two or more packets simultaneously transmitting on the channel, the only information available to the users is that a collision has occured. It may be possible to determine the multiplicity of the collision by means of an energy level detector [1-7]. This paper will show that this additional information can be used to achieve a throughput approaching one in the limit.

We will briefly describe the model next. The channel is slotted and the duration of each slot corresponds to a packet length. Packet transmissions are timed to correspond to slot durations on the channel. At the end of each slot, every user is informed whether the slot was idle (contained no transmission), had a successful transmission (contained

---

*This work was supported by the National Science Foundation under Grants NCR-8909719 and NCR-9115864, and by the New York State Foundation for Science and Technology as part of its Centers for Advanced Technology Program. This work was presented in part at the IEEE International Symposium on Information Theory, Budapest, Hungary, June 1991.

one packet), or experienced a collision of multiplicity k, $k \geq 2$ (contained k packets). The packet arrivals from all the users is assumed to be Poisson with rate λ. Throughput is defined to be the expected fraction of slots with successful transmissions. A collision resolution algorithm (CRA) is used to control retransmissions of colliding packets until they are successfully transmitted. The objective of a CRA is to maximize the throughput of the channel while maintaining stability.

It has been shown by Pippenger [1], via a non-constructive proof, that a throughput of one can be achieved. However, the best throughput achieved thus far has achieved a throughput of approximately 0.532 [2]. This CRA used a nested scheme to successfully transmit all packets in a group of colliding packets.

2 Description of the Collision Resolution Algorithm

Consider two time axes, the packet arrival time axis and the slotted channel time axis. The packet arrival time axis records the arrival of packets for transmissions from all the users. The arrival time axis is divided into equal intervals of length $\triangle$ called *epochs.* A *collision resolution interval (CRI)* will be defined as the total length of time, which in general may consist of non-contiguous time intervals, as seen on the channel and measured in channel slots, required to successfully transmit all packets that arrived in an epoch on the arrival time axis. The CRA employs the properties of a binary matrix, which we will define next:

Detecting matrices: are binary matrices $A_{m,N}$ [9], of size $m \times F_N(m)$, such that the m-tuple $\mathbf{y}$, $\mathbf{y} = A_{m,N}\mathbf{x}$, uniquely determines the N-ary $F_N(m)$-tuple $\mathbf{x}$. Here $F_N(m)$ is an integer-valued function of N and m, as defined in [9]. The procedure for constructing the matrices $A_{m,N}$ and solving for $\mathbf{x}$ given $\mathbf{y}$ is also available in [9].

We will define a family of CRA's, parameterized by m and N, and with maximum stable throughput $\lambda_{m,N}$. For each such CRA, we will choose $\triangle$ such that $\lambda_{m,N}\triangle = (N-1)/2$, where N is an integer, $N > 2$.

Let us first consider how this algorithm operates after the successful transmission of all arrivals in the interval $(0, (i-1)\triangle)$. Let the current time be CT. Then the length of the ith CRI is the number of slots the CRA requires to transmit successfully all arrivals in the interval $((i-1)\triangle, i\triangle)$. In the next slot, $(CT, CT+1)$ on the channel time axis, all the packets in the interval $((i-1)\triangle, i\triangle)$ are transmitted or *enabled.* Epochs are enabled in first-come first-served order, i.e. no interval of length $\triangle$ is enabled before the enabling of all earlier intervals on the arrival axis. Let the expected length of an CRI when an interval of length $\triangle$ contains k packet arrivals be L_k slots. We will classify all intervals into three classes, depending on the number of arrivals in the epoch.

1. **Class 1 epochs**: If the channel is idle or a successful packet is transmitted as a result of there being none or exactly one packet arrival in the ith epoch, we have successfully resolved the interval and the ith CRI ends. Thus $L_0 = L_1 = 1$.
2. **Class 2 epochs**: If a collision of multiplicity $k, k \geq N$, occurs, then use a nested scheme, as described in [2], to successfully retransmit all k colliding packets. Tight bounds on the expected length of the CRI's are available in [2], and will be used in the subsequent analysis.
3. **Class 3 epochs**: If a collision of multiplicity $k, 2 \leq k < N$, occurs then set this interval aside and enable the next interval of length $\triangle$.

Continue this process until we have set aside $F_N(m)$ Class 3 epochs. Then split each of the $F_N(m)$ epochs into two sub-intervals, as described in [2], and denote the number of arrivals in the left sub-interval of each by $\mathbf{x} = \{x_i, 1 \leq i \leq F_N(m)\}$. The CRA then performs the equivalent of a multiplication of $\mathbf{x}$ by $A_{m,N}$ as follows. Each row of the binary matrix $A_{m,N}$ indicates how an enabling is performed, where a one or zero entry is equivalent to including or leaving out of the enabling the left sub-interval of the corresponding epoch from the set of $F_N(m)$ epochs being resolved. The feedback $\mathbf{y}$ from a set of m enablings, which we will call a *pass*, will then determine, using the properties of detecting matrices, the set $\{x_i, 1 \leq i \leq F_N(m), x_i = 0, 1, 2, \ldots, N-1\}$. Since $F_N(m) > m$ for sufficiently large m, this leads to considerable savings in the number of slots needed to determine the value the x_i's. Continue using a nested procedure [2] on the two sub-intervals, until every epoch can be resolved into sub-intervals which contain either one or no arrivals. After each pass, any sub-intervals known to contain only one arrival may be enabled for successful transmission. During this process, the original $F_N(m)$ epochs will be subdivided into several smaller sub-intervals. In each pass of the detecting matrix, we need $F_N(m)$ such sub-intervals, each of which have between 2 and $N-1$ arrivals. We shall assume that sub-intervals are chosen for a pass in first-come first-served order. As these sub-intervals are resolved, there will be a time when less than $F_N(m)$ sub-intervals will be available for a pass. At this time a new epoch or epochs of length Δ will be enabled, until we can accumulate a total of $F_N(m)$ intervals with $k, 2 \leq k < N$, arrivals in each one of them. Note that if any Class 1 or 2 epochs are encountered during this process, they will be resolved as before. Thus at any arbitrary time, a new epoch is enabled only if less than $F_N(m)$ unresolved sub-intervals are available for a pass, which needs to be "filled out". Note that the number of unresolved sub-intervals generated by Class 3 epochs remains bounded and cannot exceed $F_N(m)\lfloor (N-1)/2 \rfloor$. This follows from the fact that at any time we will never have unresolved sub-intervals from more than $F_N(m)$ Class 3 epochs, and each such epoch can be sub-divided into at most $\lfloor (N-1)/2 \rfloor$ unresolved sub-intervals, since each has to have at least 2 arrivals. Figure 1 shows an example of the arrival axis, during a period when the CRA has just enabled a Class 2 epoch in an effort to generate more Class 3 epochs.

3 Stability of the CRA

For the CRA to be stable, the expected CRI length L should be less than Δ, (see e.g. [2])i.e.

$$L = E[L_k] \leq \Delta.$$

As mentioned in the last section, $L_0 = L_1 = 1$. Let us now consider Class 3 epochs, where $L_k, 2 \leq k < N$. Excluding the initial enabling of an epoch, and the k slots needed for successful transmissions, the expected number of slots needed to resove a collision of multiplicity k using the approach desribed in [2] can be bounded by $1.8784k, k \geq 2$. Excluding the initial enabling and successful transmissions, the number of passes required to resolve an epoch therefore corresponds exactly to the number of slots required in [2]. However, in this CRA, each such slot in [2] corresponds to $m/F_N(m)$ slots instead of one slot. This is due to the fact that the m slots required for a pass using the detecting matrix gives us the same information as would have $F_N(m)$ slots using the approach in [2] Therefore the corresponding expected number of slots needed can be bounded by $(m/F_N(m))1.8784k$. Thus, after taking into consideration the initial enabling and successful transmissions, $L_k < 1 + (m/F_N(m))1.8784k + k, N > k \geq 2$. For $k \geq N$, i.e. Class 2 epochs, we may use the results from [2]. Thus $L_k < 1 + 1.8784k, k \geq N$.

Thus we can define L_k^{ub}, an upper bound on $L_k, k \geq 0$, as follows

$$L_k^{ub} = 1 + (m/F_N(m))1.8784k + k, N > k \geq 0,$$

and

$$L_k^{ub} = 1 + 1.8784k, k \geq N.$$

Also, recall that $\lambda_{m,N}\Delta = (N-1)/2$. Therefore

$$L^{ub} = E[L_k^{ub}] = \sum_{k=0}^{\infty} \frac{e^{-(N-1)/2}(\frac{N-1}{2})^k}{k!} L_k^{ub}$$

$$= \sum_{k=0}^{N-1} \frac{e^{-(N-1)/2}(\frac{N-1}{2})^k}{k!}(1 + (m/F_N(m))1.8784k + k) + \sum_{k=N}^{\infty} \frac{e^{-(N-1)/2}(\frac{N-1}{2})^k}{k!}(1 + 1.8784k)$$

$$< 1 + (N-1)/2 + 0.8784 + 1.8784(m/F_N(m))(N-1)/2$$

where we have applied Chebyshev's inequality to bound the tail of a Poisson distribution. We can also lower bound $F_N(m)$ by using results from [9,10] as follows

$$F_N(m) > \frac{\frac{m+1}{2}\log(m+1) - \frac{m+1}{2}\log 4/3 - m}{\log N}$$

where all logarithms are base 2. Therefore the CRA will be stable for all values of the throughput, $\lambda_{m,N}$, that satisfy the following sufficient condition for stability:

$$L < 1.8784 + (N-1)/2 + \frac{1.8784(N-1)}{2(\frac{m \log N}{\frac{m+1}{2}\log(m+1) - \frac{m+1}{2}\log 4/3 - m})} < \Delta = (N-1)/2\lambda_{m,N}$$

which gives us

$$\lambda_{m,N} < \frac{(N-1)/2}{1.8784 + (N-1)/2 + 1.8784(N-1)/2(\frac{m \log N}{\frac{m+1}{2}\log(m+1) - \frac{m+1}{2}\log 4/3 - m})}$$

If we use very large, effectively infinite, size detecting matrices, then

$$\lim_{m\to\infty} \lambda_{m,N} < \frac{(N-1)/2}{1.8784 + (N-1)/2}$$

The expression on the right achieves a maximum of one when N (or, equivalently, Δ) is infinite. Since r.h.s $\to 1$ as $N \to \infty$, this CRA is stable as long as $\lambda < 1$. We will state this result as a theorem.

Theorem: The CRA described above for the random access channel with collisions of known multiplicity can achieve a throughput arbitrarily close to one.

This result can be extended to the Bernoulli model [1,6], where arrivals occur only at discrete time instants with probability p, by using Bernoulli probabilities instead of the Poisson distribution.

4 Concluding Remarks

We have shown a CRA that achieves a throughput approaching unity for the random access channel with collisions of known multiplicity. This is done at the cost of high delays. Further results giving the throughputs and a lower bound on the delay for the case when only finite and small collision multiplicities are detectable is reported in [11]. The CRA used here made use of "detecting" matrices, the construction of which was given by Lindström [9]. Note that the work in [1,7] used a different type of detecting matrix, which has not yet been constructed. Also, since neither of these matrices use the feedback information from each enabling, but only at the end of each pass, it may be possible to construct CRA's with lower delays than CRA's using a purely detecting matrix approach.

References

[1] N. Pippenger, "Bounds on the performance of protocols for a multiple-access broadcast channel," *IEEE Trans. on Information Theory*, vol IT-27, March 1981.

[2] L. Georgadis and P. Papantoni-Kazakos, "A Collision resolution Protocol for Random Access Channels with Energy Detectors," *IEEE Trans. on Communications*, vol COM-30, November 1982.

[3] B.S. Tsybakov, "Resolution of a conflict of known multiplicity," *Prob. Inform. Transmission*, vol 16, no. 2, translated from *Problemy Peredachi Informatsii.*

[4] M. Georgiopoulos and P. Papantoni-Kazakos, "Collision resolution protocols utilizing absorptions and collision multiplicities," *IEEE Trans. on Communications*, vol. COM-33, July 1985.

[5] N. Mehravari, " On Poisson Contention Resolution Problem with Feedback Based on Conflict Intensity," *IEEE Trans. on Communications*, vol COM-36, April 1988.

[6] N. Mehravari, " Random Multiple-Access Communication with Feedback Based on Conflict Multiplicity,"*Proc. of the 1984 Conf. on Information Sciences and Systems*, Princeton University, March 1984.

[7] Roy D. Yates, "Methods of Multiple Access Communications with Energy Detectors," M.S. Thesis, LIDS - TH - 1557, M.I.T., May 1986.

[8] Special Issue on Random-Access Communications, James L. Massey, Guest Editor, *IEEE Trans. on Information Theory*, vol. IT-31, March 1985.

[9] B. Lindström, "On a Combinatorial Problem in Number Theory", *Canadian Mathematical Bulletin*, vol. 8, no. 4, June 1965.

[10] G.F. Clements and B. Lindström, "A sequence of (±1) - determinants with large values, *Proc. Amer. Math. Soc.*, June 1965.

[11] Walter Grote, "Collision Resolution Algorithms for Random Access Channels", Ph.D. Dissertation, Department of Electrical Engineering, Polytechnic University, Brooklyn, NY, January 1992.

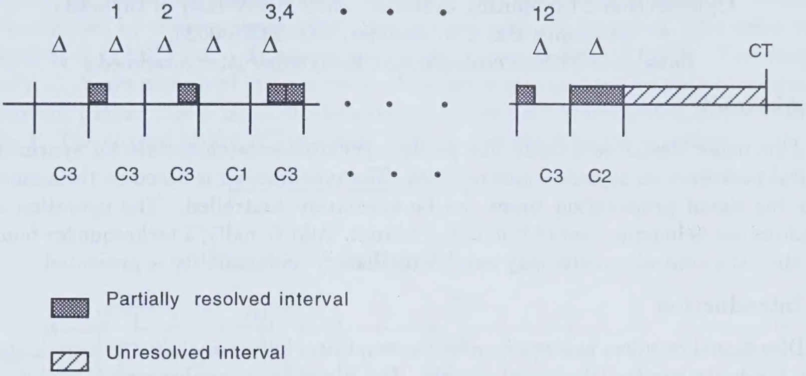

1
2
3,4
12
Δ
CT
C3
C3
C3
C1
C3
C3
C2
Partially resolved interval
Unresolved interval

Figure 1. CRA example, showing the arrival axis. C1, C2, C3 refer to Class 1,2, and 3 epochs. CT is Current Time. The CRA needs to accumulate more than the 12 intervals shown before a "pass" using a detecting matrix. The last interval enabled is a Class 2 epoch.

Format Insensitive Packet Synchronizer

Carl E. Love and Harry F. Jordan
Optoelectronic Computing Systems Center, University of Colorado,
Campus Box 525, Boulder, CO. 80309-0525.
Email love@albert.colorado.edu, harry@maruti.colorado.edu

1 Abstract

This paper describes a delay line packet synchronizer architecture for synchronizing optical packets in an asynchronous network. The synchronizer is based on the assumption that the signal propagation times can be accurately controlled. The operation of the synchronizer is independent of the packet format. Additionally, a technique for bounding the time the control circuits may exhibit oscillatory metastability is presented.

2 Introduction

Directional couplers and semiconductor amplifiers have the ability to switch and amplify terahertz bandwidth optical signals. The directional coupler cannot switch at the signal bandwidth, nor can the amplifier gain be modulated at that bandwidth. If information is encoded in long packets, these devices can be used to switch and even selectively amplify packets, provided some inter-packet gap is available for switching. Amplifying and switching are independent of the internal format of the packet. This paper discusses a way to synchronize high bandwidth packets arriving asynchronously over two independent channels in a way which is also insensitive to the internal packet format.

Such synchronization is necessary even for simple routing nodes to determine how (and when) to apply conflict resolution between overlapping packets. It is a general case of the simpler problem of synchronizing arriving packets to a local host clock. In direct networks, where link bandwidth is larger than host insertion bandwidth, compression and decompression can be done by a host to process the internal structure of buffered packets, but routing must be done at the full link speed.

The synchronizer presented here depends only on the packet envelope. If optical power is guaranteed to be present at the start of a packet - start bit - and the packet duration is known precisely, such an envelope signal can be generated by using a delayed version of the start bit to mark the end. The start bit leading edge must be resolved precisely, the rest of the packet structure is not needed to generate the envelope signal.

The synchronizer is implemented with 2x2 optical switches and fiber or integrated waveguide delay lines. Other technologies are conceivable, but two assumptions must be satisfied: rise and fall times must be short compared to circuit and device delays, and delays must be precisely controllable. Optical technology can satisfy these assumptions, while current electronic techniques are hard to fit into this mold.

Another technique for optical packet synchronization involves extraction of clock from the packet using a phased locked loop [1], [2], and [3]. In addition to examining the internal packet structure, this technique requires a very fast clock. The autocorrelation techniques in [4], [5] may be able to support the start bit recognition needed here.

3 Synchronization System Overview

The format insensitive packet synchronizer is based on the reference delay line, which is similar to other delay line architectures found in the literature [5], [6]. The architecture can be used to align two packets by advancing or delaying one of the packets with respect to the other packet until they are aligned. The packet separation T is assumed to be equal to the fixed packet length, T_L, plus the minimum gap between packets, $S_{min} > 0$, as shown in Fig. 1. The delay line, shown in Fig. 2, is one of two possible implementations (see [6] for the other implementation). The delay line consists of a series of switching nodes connected by three optical paths: 1) a path of constant length d_k, 2) a delay path of length $d_k + (1/2)^k T$, and 3) an advance path of length $d_k - (1/2)^k T$. The lengths $d_k, k = 1, ..., N$ are measured in time units. A single continuous path exists consisting of the constant paths. The α input to the constant delay path is connected to the α input of node 1. The output of the N^{th} constant path is connected to the α output.

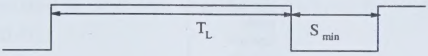


Figure 1: Definition of S_{min} and T_L.

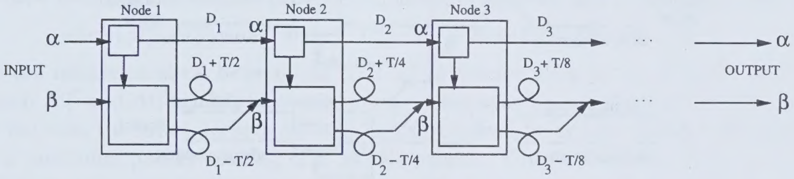


Figure 2: Reference Delay Line Architecture.

Packets travel in the same direction. The packet on the adjustable input to a switching node is routed to the delayed output if it arrives before the packet on the constant path. Otherwise, it is routed to the advance output. The maximum skew between packets being aligned is halved at each successive node.

The leading edge times correspond to the arrival time of the packet envelope signal as shown in Fig. 3. The packet envelope times are used to determine the packet routings, independent of the internal packet structure. The list of leading edge times for the α input stream is denoted by $L_i^\alpha, i = 1, 2, 3, ..., N_A$ where $N_A \geq 1$. The α leading edge times are assumed to meet the following conditions:

$$0 \leq L_i^\alpha < L_{i+1}^\alpha, \quad \forall i\,, \text{ and } T_L + S_{min} \leq L_{i+1}^\alpha - L_i^\alpha, \quad \forall i\,. \tag{1}$$

The β input leading edge times is similarly defined by L_j^β, for $j = 1, 2, 3, ..., N_B$, $N_B \geq 1$. The β leading edge times must also meet condition 1 with β substituted for α and j substituted for i. The input packets are mapped to either M_i^α, $i = 1, .., M_A$ or M_j^β, $j = 1, .., M_B$ such that $M_A + M_B = N_A + N_B$. The M_i^α times must meet the condition:

$$0 < M_i^\alpha < M_{i+1}^\alpha, \quad \forall i\,, \text{ and } T_L + S_{min} < M_{i+1}^\alpha - M_i^\alpha, \quad \forall i. \tag{2}$$

Output list M_j^β is the list of leading edge times for the β output stream which must also meet condition 2 with β substituted for α and j substituted for i. The two output streams are properly synchronized if they meet one of the following conditions:

$$\mid M_i^\alpha - M_j^\beta \mid \geq T_L + S_{min}\,, \text{ or } \mid M_i^\alpha - M_j^\beta \mid \leq \epsilon, \ \forall (i, j). \tag{3}$$

Ideally $\epsilon = 0$, however in a real system $\epsilon \geq T_L (1/2)^{N+1}$ for an N node line. Equation 3 states the packets are either synchronized or the packets do not overlap.

3.1 Reference Delay Line. The routing of the packet on the adjustable input path to either the advance or the delayed output is done by a 2x2 crossbar switch as shown in Fig. 4. The arrival times of packets at the inputs of the k^{th} node in the reference delay line are defined by the list of ordered leading edge times, $L_i^{c,k}, i = 1, ..., N_C$ and $L_j^{a,k}, j = 1, ..., N_A$, where $L_i^{c,k}$ are the arrival times for the constant path and $L_j^{a,k}$ are the

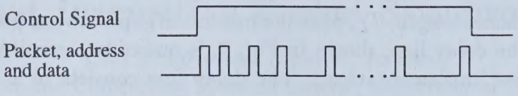

Control Signal
Packet, address and data

Figure 3: Control Pulse.

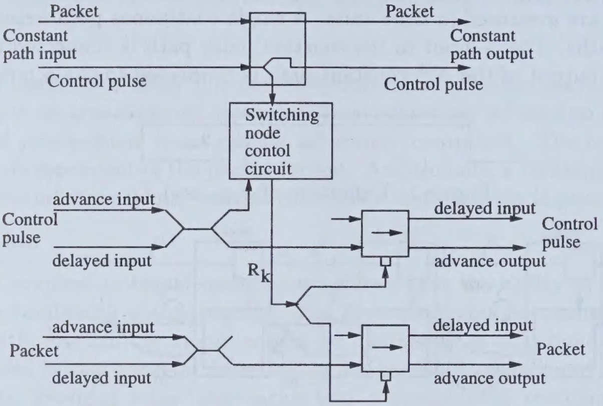

Packet
Constant path input
Control pulse
Packet
Constant path output
Control pulse
Switching node contol circuit
advance input
Control pulse
delayed input
R_k
delayed input
Control pulse
advance output
advance input
Packet
delayed input
delayed input
Packet
advance output

Figure 4: Detailed Reference Delay Line Switching Node Diagram.

arrival times for the adjustable path to the k^{th} switching node. Let D, E, and F be a set of Boolean variables representing the relationships between the two lists of leading edge times. The definitions of D, E, and F are:

$$D(i,j,k) = (L_j^{a,k} \geq L_i^{c,k}), \quad E(i,j,k) = (0 \leq L_j^{a,k} - L_i^{c,k} \leq T(1/2)^k), \text{ and} \tag{4}$$

$$F(i,j,k) = (0 < L_i^{c,k} - L_j^{a,k} \leq T(1/2)^k). \tag{5}$$

The setting of the 2x2 crossbar routing switch at node k is given by the Boolean variable X_k ($X_k = 1$ bar state, $X_k = 0$ cross state). The value of X_k at the k^{th} node in the reference delay lines is given by the following hardware level algorithm.

```
$i = 1$;   $j = 1$;
Until $\overline{(i + j > N_A + N_C)}$ then {
    if $(D)$ then {
        $X_k = 0$;
        if $(E)$ then $i = i + 1$;
        $j = j + 1$;
    } else {
        $X_k = 1$;
        if $(F)$ then $j = j + 1$;
        $i = i + 1$;
    }
}.
```

The reduction in skew between packets at successive nodes is characterized by the following theorem. The proof of the theorem can be found in [6].

> **Theorem.** Given that the difference in the arrival times at node k of two packets is bounded between 0 and $(1/2)^{k-1}T$, the difference in the arrival times of the two packets at node $k+1$ is bounded between 0 and $(1/2)^k T$. T is the maximum difference in the arrival times of two packets that can aligned by the reference delay line.

The α packet spacing does not change from input to output. If a packet enters on the β input and there is no packet to align it to on the α input, the minimum packet spacing for the β packets cannot be guaranteed at the output. In general, the spacing of the packets at the output is not guaranteed to meet the minimum packet spacing requirement for two arbitrary input streams, as can be seen in the following example. Consider an the input sequence defined by the conditions:

$$T < L_1^\beta - L_1^\alpha, \text{ and } 0 < \mathrm{L}_2^\alpha - \mathrm{L}_2^\beta < \mathrm{T}, \tag{6}$$

shown in Figure 5. This is a case in which the first α and β packets do not overlap another packet. The second α and the second β packets do overlap. The output times of the packets are given by

$$M_1^\alpha = L_1^\alpha + d_T, \quad M_1^\beta = L_1^\beta + d_T + T, \quad M_2^\alpha = L_2^\alpha + d_T, \text{ and } \quad M_2^\beta = L_2^\alpha + d_T. \tag{7}$$

The skew between the aligned packets (L_2^α and L_2^β) and the first β output packet is

$$skew(M_1^\beta, M_2^\beta) = L_2^\alpha + d_T - L_1^\beta - d_T - T = L_2^\alpha - L_1^\beta - T > 0, \tag{8}$$

since the minimum skew between L_1^β and L_2^α is greater than or equal to T. The skew between M_1^β and M_2^β violates the minimum packet spacing requirement. In general, the skew between the input packets on each input line must be at least $2T$ in order to ensure that a minimum packet spacing of T at the output. Since the minimum input skew of the β packets is greater than the minimum output skew the reference delay line is not cascadable for all possible input sequences.

3.2 Requirements for Cascadable Synchronizer. For a synchronizer to be cascadable, the minimum spacing of the packets at the output must be greater than or equal to the minimum input spacing for all possible input sequences. The reference delay line synchronizer can be made cascadable if packets not having another packet with which to align them are always routed to the constant delay path of the reference delay line synchronizer. The mapping of the input times L^α and L^β to the output times M^α and M^β can be defined to guarantee the synchronizers are cascadable. The following boolean conditions are used in the cascadable synchronization algorithm:

$$G(i,j) = (L_i^\alpha \leq L_j^\beta), \qquad R(i,j) = (L_i^\alpha \leq L_j^\beta \leq L_i^\alpha + T_L + S_{min}), \tag{9}$$

$$S(i,j) = (L_j^\beta < L_i^\alpha \leq L_j^\beta + T_L + S_{min}), \qquad T(i,j) = (1 \leq i \leq N_A), \text{and} \tag{10}$$

$$U(i,j) = (1 \leq j \leq N_B). \tag{11}$$

The cascadable synchronization algorithm that specifies the desired input to output mappings is given by the following pseudo code, where F_d is a frame delay for the synchronizer.

```
$i = 1$;   $j = 1$;
Until $(\overline{T + U})$ do {
        if $(G)$ then {
                $M_i^\alpha = L_i^\alpha + F_d$;
                if $(R \cdot U)$ then { $M_j^\beta = M_i^\alpha$;   $j = j + 1$; }
                $i = i + 1$;
        } else {
                $M_j^\beta = L_j^\beta + F_d$;
                if $(S \cdot T)$ then { $M_i^\alpha = M_j^\beta$;   $i = i + 1$; }
                $j = j + 1$;
                }
        }.
```

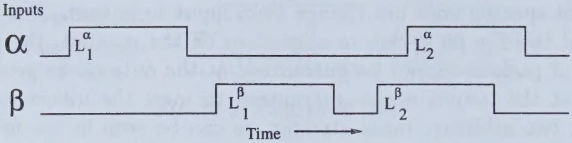


Figure 5: A single packet followed by two overlapping packets.

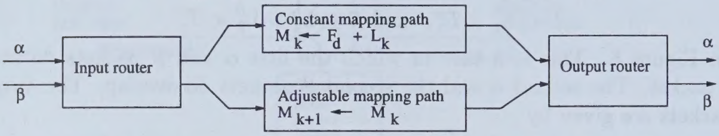


Figure 6: Two-Stream Synchronizer Architecture.

The algorithm ensures that the conditions of Eqn. 2 hold and that every input time is mapped to a unique output time, as proven in [6].

3.3 Cascadable Synchronization Architecture. The synchronization algorithm can be implemented by the architecture of Fig. 6. The next leading edge time L_i^α or L_j^β is routed to the constant delay path by the input router. Input routing is based on conditions $G(i,j,k)$, $R(i,j,k)$, $S(i,j,k)$, $T(i,j,k)$, and $U(i,j,k)$. The reference delay line maps from L_i^α and L_j^β to M_i^α and M_j^β. The output router ensures that the packets exit on the line corresponding to the input line on which they arrived.

4 Synchronizer Architecture Implementation

The input router determines the input on which the next leading edge arrives and whether the arrival time of the next leading edge (on the other input) is within $T_L + S_{min}$ of the prior edge. These decisions correspond to the determination of conditions G, R, and S. The routing of the packets is done by 2x2 crossbar switches. The switches are controlled by sequential circuits, which have the potential of metastable operation. A technique called logic signal stretching is introduced as a technique for bounding the time a circuit may exhibit oscillatory metastability. Finally, the design of control circuits for the input router and the reference delay line is discussed.

4.1 Router Design. The design assumes that $S_{min} < T_L$. The input router sends the input packets to one of two mapping paths based on the conditions $G(i,j)$, $R(i,j)$, and $S(i,j)$ of the algorithm. The output router routes packets from the two mapping paths to the output corresponding to the input on which they arrived. The packet consists of a destination address and data. In principle, the packet information can be encoded by power, wavelength, or phase. The following discussion assumes that the packet information is power encoded and a control pulse is associated with each packet.

A block diagram of the input router is given in Fig. 7 that shows the three inputs to the control circuit P, Q, and V. It also shows two control outputs Z_α and Z_β. Signal P is a logic 1 if there is a packet at the α input to the input router and a logic 0 otherwise. Q is similarly defined for the β input to the router. Condition $G(i,j)$ is determined in hardware by the order in which P and Q change from logic 0 to logic 1. It is assumed that signals P and Q are both logic 0 initially. An input sequence causing signal P to change to a logic 1 before signal Q changes to a logic 1 corresponds to condition $G(i,j)$ being *True*. Condition $G(i,j)$ is *False* if Q changes to a logic 1 before signal P changes.

Conditions $R(i,j)$ and $S(i,j)$ in the two stream algorithm were defined in Eqns. 9 and 10. One of the conditions is *True* if the second packet arrives within $T_L + S_{min}$ of the

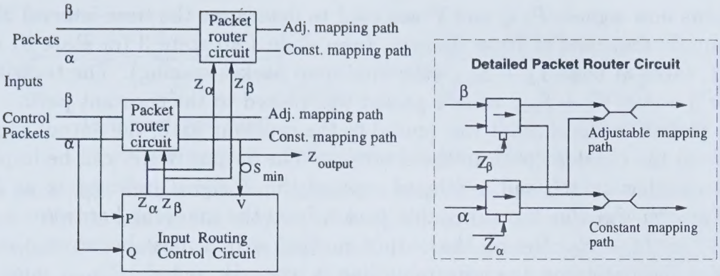

β
Packets
α
Inputs
Packet router circuit
Adj. mapping path
Const. mapping path
Z_α
Z_β
β
Control Packets
α
Packet router circuit
Adj. mapping path
Const. mapping path
Z_{output}
S_{min}
Z_α Z_β
V
P
Q
Input Routing Control Circuit
Detailed Packet Router Circuit
β
Adjustable mapping path
Z_β
α
Constant mapping path
Z_α

Figure 7: Detailed Architecture of the Input Router.

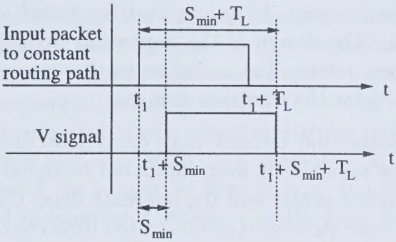

$S_{min} + T_L$
Input packet to constant routing path
t
t_1
$t_1 + T_L$
V signal
t
$t_1 + S_{min}$
$t_1 + S_{min} + T_L$
S_{min}

Figure 8: Determination of time interval $T_L + S_{min}$.

first packet. Signal V in Fig. 7 is used by the hardware to determine the time interval for the two conditions. Signal V is a copy of the control pulse that was routed to the constant delay line. Signal V is delayed by S_{min} to make its falling edge occur $T_L + S_{min}$ after the rising edge of the packet on the constant path, as shown in Fig. 8.

Output signals Z_α and Z_β set the 2x2 switches to route the α and β packets as shown in Fig. 7. They are defined by the equations $Z_\alpha = Z \cdot P$ and $Z_\beta = Z \cdot Q$, where Z corresponds to condition $G(i,j)$. Z is 1 ($G(i,j)$ is $True$) if the α input is to be connected to the constant path and 0 if the if the α input is to be connected to the adjustable path. The definition of Z is given by the state diagram for the input routing control circuit in Fig. 9. If $Z_\alpha = Z_\beta$, both the α and the β packets are routed to different mapping paths assuming that both packets are present, i.e. $P = 1$ and $Q = 1$, so no routing conflicts can occur. Z_α is a *don't care* if $P = 0$ since no α packet is present, and Z_β is a don't care if $Q = 0$. If only one packet is present no routing conflict can occur. Thus it is seen that Z_α and Z_β properly route the packets on the basis of P, Q, V, and Z. Signals Z_α and Z_β do not change until the routing of the packets is completed. The state diagram in

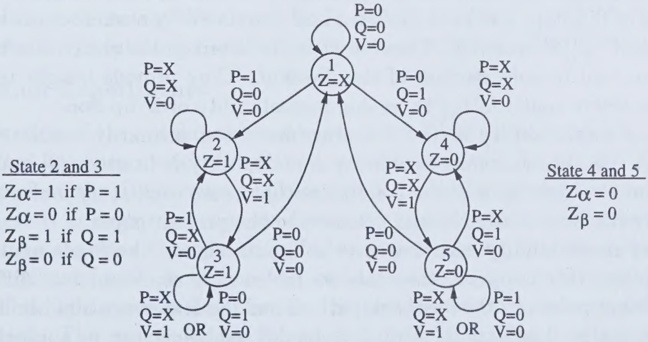

P=0 Q=0 V=0
1 Z=X
P=X Q=X V=0
P=1 Q=0 V=0
P=0 Q=1 V=0
P=X Q=X V=0
2 Z=1
P=X Q=X V=1
4 Z=0
P=X Q=X V=1
State 2 and 3
$Z_\alpha = 1$ if $P = 1$
$Z_\alpha = 0$ if $P = 0$
$Z_\beta = 1$ if $Q = 1$
$Z_\beta = 0$ if $Q = 0$
State 4 and 5
$Z_\alpha = 0$
$Z_\beta = 0$
P=1 Q=X V=0
P=0 Q=0 V=0
P=0 Q=0 V=0
P=X Q=1 V=0
3 Z=1
5 Z=0
P=X Q=X V=1 OR P=0 Q=1 V=0
P=X Q=X V=1 OR P=1 Q=0 V=0

Figure 9: State Diagram for The Input Router.

Fig. 9 shows how signals P, Q and V are used to determine the time interval $T_L + S_{min}$. The minimum sequence of state changes, from state 1 to state 3 (or state 5) and back to state 1, takes at least $T_L + S_{min}$ (the minimum packet spacing). The transition from state 3 or 5 occurs $T_L + S_{min}$ after a packet was routed to the constant path.

If the packet on the α input was routed to the constant path, the output router must route it from the constant path to the α output. The output router can be implemented by a 2x2 crossbar switch and a delayed copy of the Z signal, referred to as Z_{output} in Fig. 7. The exit time for the adjustable packet from the mapping hardware is given by $M_j^{\beta} = M_i^{\alpha} = L_i^{\alpha} + F_d$. Hence, the output routing switch must be set at time M_i^{α} to ensure that the packet on the adjustable line is properly routed. Z_{output} must be valid until $M_i^{\alpha} + T_L$ to ensure that the packets are properly routed. The frame delay, F_d, is equal to the length of the constant delay line plus the length of the paths through the input and output routers. The design of the sequential circuits for routing the packets at each node in the reference delay line is similar to the design of the sequential circuit for the input router, see [6] for the complete designs.

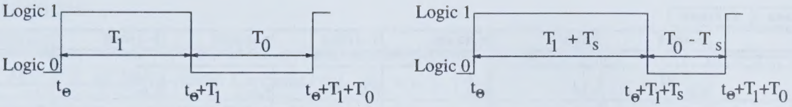
4.2 Metastability. Sequential circuits may exhibit oscillatory metastability if the feedback time of the state variables is long compared to signal rise and fall times. The sequential circuits in the input router and the reference delay line may exhibit oscillatory metastability. The use of logic stretching to bound the time the circuits exhibit oscillatory metastability is introduced in this section.

In the early 1970's it was noticed that many computer systems were encountering significant failure rates due to synchronizer failures between asynchronous subsystems [7]. This phenomenon has been termed anomalous behavior of synchronizers and arbiter circuits [8]. It is the result of one or more flip-flops entering a metastable state of operation. Two types of metastable behavior have been identified. The first type, called analog metastability [9], is characterized by the output of the flip-flop maintaining an output voltage outside of the defined voltage range for a logic 0 or a logic 1. The second is known as oscillatory metastability [7] and [10]. Oscillatory metastability typically occurs in flip-flops having a long propagation time (or internal feedback time) in comparison to the rise and fall times of the inputs. The optical implementaton of the circuits in Section 4.1 have this property.

Circuit designs have been proposed that claim to reduce or eliminate analog metastable operation [8], [11], and [12]. The basic premise of these designs has been to make the switching of the circuit from one stable state to another very fast relative to the changes of their inputs. These techniques have reduced the susceptibility of the circuit to analog metastable operation, but they cannot guarantee the elimination of metastability [13]. The behavior of flip-flops has been modeled and simulated by numerous authors [9], [11], [14], [15], [16], [17], [18], and [19]. These models are based on the electrical characteristics of the devices used in construction of the flip-flop. They provide insight into the wide range of parameters contributing to analog metastability of a flip-flop.

Analysis of metastability in the literature has been primarily restricted to analog metastability. The literature on oscillatory metastability is limited ([9] and [18]). The analysis of and the discussion of the conditions that cause oscillatory metastability in [9] is particularly relevant to the circuits discused in the present paper.

Oscillatory metastability occurs when the inputs violate the fundamental mode assumption. When this happens, the state variables may represent two different stable states at different points in the feedback path. Thus the state variables oscillate between the two stable states. Logic signal stretch, to be defined below, can be applied to the state



Unstretched input signal Logic 1 signal stretched by T

Figure 10: Definition of Logic 1 Stretch

variables such that each oscillation causes the circuit to spend more time in one of the two states. Within a bounded number of oscillations, the time spent in the chosen state is made equal to the state variable feedback time thus stabilizing the circuit. Oscillatory metastability is discussed in terms of transitions in a state table in [6]. Additionally, presented in [6] is an algorithm to determine the set of state variables requiring logic 1 stretch and logic 0 stretch to bound the time the circuit exhibits oscillatory metastability. The two sets of state variables must be disjoint. The sequential circuits designed for the synchronizer have disjoint sets. However, it is not known in general if a state assignment exists for an arbitrary sequential circuit such that the set of variables requiring logic 1 and logic 0 stretch will be disjoint. Logic signal stretching requires the accurate control of signal propagation times that is possible in a time-of-flight circuit. The properties of stretch are a combination of the properties seen in pure and inertial delays described in [20]. Stretch can be used to lengthen the logic 1 or the logic 0 state of a signal.

> **Definition: Logic 1 stretch.** Let Y be a logic signal that is a logic 1 from time t_θ to time $t_\theta + T_1$ and is then a logic 0 from time $t_\theta + T_1$ to time $t_\theta + T_1 + T_0$, as shown in Fig. 10. If the logic 1 value is stretched by T_s, then Y maintains a logic 1 value from time t_θ to $t_\theta + T_1 + T_s$ and a logic 0 from time $t_\theta + T_1 + T_s$ to time $t_\theta + T_1 + T_0$. The position of the logic 0 to logic 1 signal transition at time t_θ does not change when logic 1 stretching is applied, assuming the previous logic 1 to logic 0 transition occurred more than T_s earlier. If the previous logic 1 to logic 0 transition did not occur more than T_s earlier, the signal will be a logic 1 before and after time t_θ. The logic 1 to logic 0 transition moves from time $t_\theta + T_1$ to $t_\theta + T_1 + T_s$ if $T_0 > T_s$; the transition disappears if $T_0 \leq T_s$.

Logic 1 stretch adds a pure delay to the logic 1 to logic 0 transition without adding any delay to the logic 0 to logic 1 transition . Secondly, logic 1 stretch removes logic 0 pulses whose duration T_0 is less than T_s. The logic 1 pulse width T_1 is increased by T_s provided $T_0 > T_s$. Stretch is an asymmetric process whereas pure delays and inertial delays are both symmetric processes. Logic 1 stretch can added to a signal by ORing the signal and a copy of the signal delayed by T_s, see [6] for more details. Logic 0 stretch is similarly defined by replacing T_0 with T_1 and logic 0 with logic 1 in the above definition.

5 Simulation Experiments

The operation of the synchronizer was simulated using a discrete event optical circuit simulator (2x2 crossbar switches, splitters, combiners, and fiber) called XHatch which was developed at the University of Colorado [22]. Although exact simulation is not possible, there is enough resolution to demonstrate the behavior described above. The simulation demonstrates the operation for an input stream that contains overlapping packets to be aligned with each other and input packets with no overlapping packet. The optical time-of-flight circuits described in the previous section were used in the 5 node line simulation. The packet length T_L equaled 5760 simulation time units (STU). The

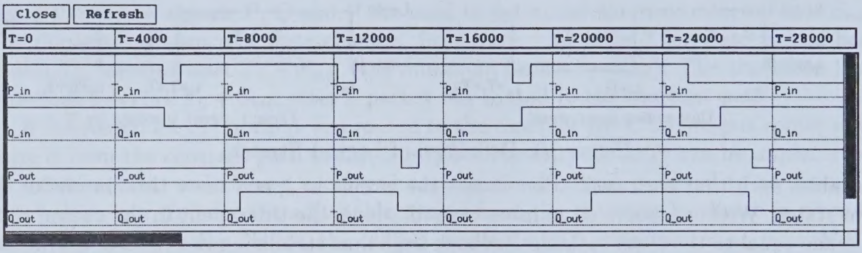

Close
Refresh
T=0
T=4000
T=8000
T=12000
T=16000
T=20000
T=24000
T=28000
P_in
Q_in
P_out
Q_out

Figure 11: Two-stream Synchronizer Output, 1.

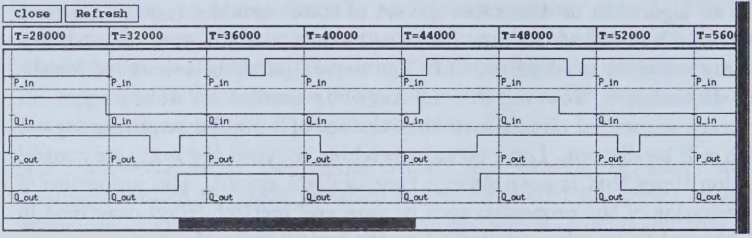

Close
Refresh
T=28000
T=32000
T=36000
T=40000
T=44000
T=48000
T=52000
T=560
P_in
Q_in
P_out
Q_out

Figure 12: Two-stream Synchronizer Output, 2.

minimum packet separation S_{min} equaled 640 STU. The alignment tolerance was $T_L/16$. The feedback time was $T_f = 10$ STU. The length of the line was 8400 STU.

The simulated input sequence includes packets whose difference in arrival times range from a small amount to amounts greater than T. In each case, the synchronizer correctly aligns the packets. The α and β inputs to the synchronizer are given by P_{in} and Q_{in} respectively on the XHatch graphs. The α and β outputs from the synchronizer are given by P_{out} and Q_{out} respectively. Figures 11 through 13 show the results of a simulation experiment. The second, fourth, sixth, eight, ninth and tenth P packets do not overlap a Q packet. The first, third, fifth, and seventh P packets overlap one of the first four Q packets. The final Q packet is a single packet.

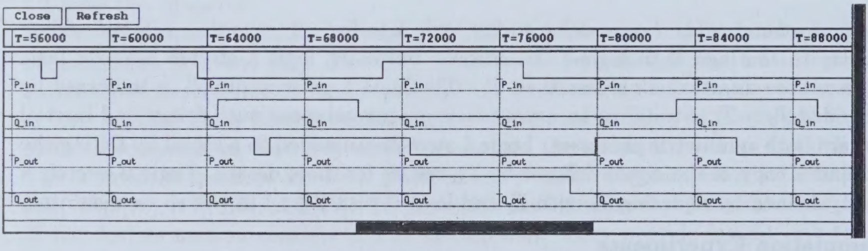

Close
Refresh
T=56000
T=60000
T=64000
T=68000
T=72000
T=76000
T=80000
T=84000
T=88000
P_in
Q_in
P_out
Q_out

Figure 13: Two-stream Synchronizer Output, 3.

6 Conclusion

An all optical packet synchronizer for use in high speed computer and communication networks was presented. The synchronizer does not examine the internal structure of packets to extract timing information. It is based on the assumption that signal propagation times can be controlled accurately. A technique called logic signal stretching was introduced to bound the time that a sequential circuit may exhibit oscillatory metastability. Finally, simulation results were presented to demonstrate its operation.

References

[1] B. Kim, et al., "A 30-MHz hybrid analog/digital clock recovery circuit in 2-μm cmos," *IEEE J. of Solid-State Circuits*, vol. 25, no. 6, pp. 1385–1394, Dec. 1990.

[2] J. F. Davey, "Modems," *Proc. of the IEEE*, vol. 60, pp. 1284–1292, Nov. 1972.

[3] B. Lathi, *Modern Digital and Analog Communication Sys.*. CBS College Pub., 1983.

[4] P. R. Prucnal, et al., "Demonstration of a rapidly tunable optical time-division multiple-access coder," *IEEE Photonics Tech. Letters*, vol. 3, pp. 170–2, Feb. 1991.

[5] P. R. Prucnal, "Optically processed self-routing, synchronization, and contention resolution for 1-d and 2-d photonic switching architectures," *IEEE J. of Quantum Electronics*, vol. 29, no. 2, pp. 600–611, Feb. 1993.

[6] C. E. Love, *Optical Synchronization Techniques.* PhD thesis, Optical Computing Systems Center, University of Colorado, Boulder, CO 80309-0525, 1995.

[7] T. J. Chaney and C. E. Molnar, "Anomalous behavior of synchronizer and arbiter circuits," *IEEE Trans. on Computers*, vol. 22, pp. 421–422, April 1973.

[8] G. Elineau and W. Wiesbeck, "A new J-K flip-flop for synchronizers," *IEEE Trans. on Computers (Correspondence)*, vol. 26, pp. 1277–1279, Dec. 1977.

[9] L. Reyneri, et al., "Oscillatory metastability in homogeneous and inhomogeneous Flip-Flops," *IEEE J. of Solid-State Circuits*, vol. 25, pp. 254–264, Feb. 1990.

[10] J. U. Horstmann, et al., "Metastability behavior of CMOS ASIC flip-flops in theory and test," *IEEE J. of Solid-State Circuits*, vol. 24, pp. 146–157, Feb. 1989.

[11] L.-S. Kim, "Metastability of CMOS latch/flip-flop," *IEEE J. of Solid-State Circuits*, vol. 25, pp. 942–951, Aug. 1990.

[12] E. G. Wormald, "A note on synchronizer or interlock maloperation," *IEEE Trans. on Computers*, vol. C-26, pp. 317–318, March 1977.

[13] T. Chaney, "Comments on 'a note on synchronizer or interlock maloperation'," *IEEE Trans. on Computers*, vol. c-28, pp. 802–804, Oct. 1979.

[14] M. Hurtado and D. L. Elliott, "Ambiguous behavior of logic bistable systems," in *Proc. of the 13th Allerton Conf. on Circuit Theory*, Oct. 1975, pp. 605–611.

[15] L. Kleeman, "The jitter model for metastability and its application to redundant synchronizers," *IEEE Trans. on Computers*, vol. 39, pp. 930–942, July 1990.

[16] G. Couranz and D. Wann, "Theoretical and experimental behavior of synchronizers operating in the metastable region," *IEEE Trans. on Comp.*, vol. c-24, pp. 604–616, June 1975.

[17] J. H. Hohl, et al., "Prediction of error probabilities for integrated digital synchronizers," *IEEE J. of Solid-State Circuits*, vol. SC-19, no. 2, pp. 236–244, 1984.

[18] T. Kacprzak, "Analysis of oscillatory metastable operation of an RS flip-flop," *IEEE J. of Solid-State Circuits*, vol. 23, pp. 260–266, Feb. 1988.

[19] L. R. Marino, "General theory of metastable operation," *IEEE Trans. on Computers*, vol. c-30, pp. 107–115, Feb. 1981.

[20] S. H. Unger, *Asynchronous Sequential Switching Circuits.* John Wiley & Sons, 1969.

[21] J. Pratt and V. Heuring, "Delay synchronization in time-of-flight optical systems," *Applied Optics*, vol. 31, pp. 2430–2437, May 10 1992.

[22] M. Salerno, "Xhatch," OCS TR91-25, University of Colorado, Optoelectronic Computing Systems Center, University of Colorado, Boulder, CO 80309-0425, 1991.

Routing Schemes for Multiple Random Broadcasts in Arbitrary Network Topologies[1]

Emmanouel (Manos) Varvarigos and Ayan Banerjee
University of California - Santa Barbara
Department of Electrical and Computer Engineering
Santa Barbara, CA 93106
E-mail: manos@ece.ucsb.edu

Abstract

Broadcasting is the operation where a packet is copied from a node to all the other nodes of a network. In this paper we assume that packets are generated continuously, over an infinite time horizon with rate λ, and each of them has to be broadcast to all the other nodes of the network. The network topology is assumed to be an arbitrary bidirectional graph. We derive upper bounds on the maximum achievable broadcast throughput, and lower bounds on the average time required to complete a broadcast. These bounds apply to any network topology, independently of the scheme used to perform the broadcasts. We also propose two dynamic broadcasting schemes, called the indirect and the direct broadcasting scheme, that can be used in a general topology, and we evaluate analytically their throughput and average delay. The throughput achieved by the proposed schemes is equal to the maximum possible, if a half-duplex link model is assumed, and is at least equal to one half of the maximum possible, if a full-duplex model is assumed. The average delay of both schemes is of the order of the diameter of the trees used to perform the broadcasts. The analytical results obtained do not use any approximating assumptions.

1. INTRODUCTION

We focus on the problem where broadcast requests are generated at random time instants at each node of a multiprocessor network that has an arbitrary topology. In particular, we assume that packets are generated at each node according to a Poisson process with rate λ, and that each of them has to be broadcast to all other nodes. We are interested in finding efficient routing schemes to execute the broadcasts in a general topology, and in evaluating their performance. The assumption of Poisson arrivals is made anly because the mathematics of the analysis require it and it is inessential for the implementation of the schemes that we propose. The dynamic broadcasting problem arises in many applications, and we believe it deserves a position among the generic network routing problems.

The dynamic broadcasting problem was first addressed in [StT93] for hypercubes, and in [VaB95] and [VaB94a] for hypercubes and d-dimensional meshes, respectively. The routing schemes and analytical results that we develop in the present paper are considerably more general, since they apply to arbitrary network topologies, that may or may not have any symmetry properties. The throughput and delay for unicast (one-to-one) communication have been examined extensively in the litterature for a number of topologies (see, for example, [GrG86], [GrH92], [Kri90], [VaB94b]), with the analysis being approximate, except for the results given by Stamoulis and Tsitsiklis in [StT94] for hypercubes and butterflies, which did not use any approximating assumptions. We believe that the throughput and the delay of a network for broadcast (one-to-many) communication are equally important criteria in evaluating network performance. The analysis to be presented on the dynamic broadcasting problem will not use any independence or other approximating assumptions. We consider this particularly important, since the analysis of problems that involve networks of queues is in general extremely difficult. For example, no accurate analysis exists, to the best of our

---

[1] Research supported under contracts by ARPA under Contract DABT63-93-C-0039, and by NSF under Grant NSF-RIA-08930554.

knowledge, for the throughput and delay of the corresponding unicast communication problem in a general network topology, despite the efforts of many researchers.

We assume that store-and-forward switching is used for the transfer of data. All packets have equal length and they require one unit of time (or slot) in order to be transmitted over a link. We consider two communication models. In the first model, called Multiple Link Availablity Full Duplex model (or F-D model, for brevity), a node is capable of transmitting and receiving messages along all its incident links concurrently, and each link can be used for transmission along both directions simultaneously. In the second model, called Multiple Link Availablity Half Duplex model (or H-D model, for brevity), a node can use all its incident links simultaneously, but a link can be used for transmission along only one direction at a time.

We are interested in three performance measures for the performance of the dynamic broadcasting scheme that we propose. The first is the maximum broadcast throughput, which is the maximum generation rate λ per node that can be accommodated by a broadcasting scheme. The second is the average broadcast delay $\mathcal{B}$, which is the average time that elapses between the generation of a packet at a node and the time its broadcast to all the other nodes is completed. The third performance measure is the average reception delay $\mathcal{R}$, which is the average time that elapses between the generation of a packet at a node and the time a particular node s receives a copy of the packet, averaged over all nodes s of the network. Since for the broadcast of a packet to be completed all nodes must receive a copy of it, we have $\mathcal{R} \leq \mathcal{B}$, for any broadcasting scheme. The average reception delay is important in the case where a processor can start processing a packet as soon as it receives it, without having to wait for the packet to be delivered to all the other processors.

We show that a necessary stability condition under the F-D model is $\lambda < \min\{2k_{\max}/N, d_{\min}/(N-1)\}$, where $k_{\max}$ is the maximum number of edge-disjoint spanning trees, $d_{\min}$ is the minimum node degree, and N is the number of nodes of the network. When the H-D model is assumed, the corresponding bound on the throughput is $\lambda < k_{\max}/N$. We also give lower bounds on the average broadcast delay $\mathcal{B}$ in terms of the parameters $k_{\max}$ and $d_{\min}$. The bounds derived hold for *any* network, and for *any* scheme that can be used to perform the broadcasts. We introduce two new routing schemes, called the *indirect broadcasting* and the *direct broadcasting* schemes, which can be used to perform the broadcasts in an arbitrary bidirectional graph G. We evaluate the throughput of the schemes under the F-D and the H-D model, and we compare it to the corresponding upper bounds. We also obtain analytical expressions for the average broadcast delay and the average reception delay of the schemes.

The indirect broadcasting scheme is proved to be stable for $\lambda < k_{\max}/N$, under both the F-D and the H-D model. Therefore, the maximum throughput achieved by the indirect scheme for a given network under the H-D model is equal to the maximum possible for that model. When the F-D model is assumed, the throughput of the indirect scheme is between 0.5 and 1 of the maximum possible (given by the upper bound). We also show that, under the F-D model, this is the best stability region that we could expect for an algorithm that works for *general* topologies (that is, for some - but not all - topologies, $\lambda < k_{\max}/N$ is also a necessary stability condition under the F-D model). We also evaluate the average broadcast delay of the indirect broadcasting scheme, and show that it is of the order of the average diameter of the spanning trees used by the scheme (or of the order of the maximum diameter of the spanning trees used, if a different version is used), for any load in the stability region. In the course of developing the indirect broadcasting scheme, we introduce a new communication task, called the *Generalized Multinode Broadcast*, and propose efficient algorithms to execute it in a general topology. The Generalized Multinode Broadcast, in addition to being an important component of the direct broadcasting scheme, it is also important on its own merit, since it arises in a number of other applications.

The second broadcasting scheme that we consider is the direct broadcasting scheme, which is a particularly simple scheme to implement. We analyze its throughput under the F-D model, and prove that the scheme is stable for $\lambda < k_{\max}/N$. We also obtain an expression for the average reception delay of the scheme and show that it is of the order of the average mean internodal distance of the spanning trees used, for any load in the stability region.

2. NOTATION AND PRELIMINARY RESULTS

The network topology is given by a general (bidirectional) graph $G = (\mathcal{N}, \mathcal{E})$. Each pair $e = [s,t]$ of $\mathcal{E}$ is called an *edge*, and corresponds to a bidirectional communication link between processors s and t. The cardinality of the sets $\mathcal{N}$ and $\mathcal{E}$ is denoted by N and E, respectively. An edge $[s,t]$ of G consists of the two unidirectional links (s,t) and (t,s) to be referred to as *arcs*. The ***diameter*** δ of a graph is defined as the maximum shortest distance between any two nodes of the network. A ***spanning tree*** is a subgraph of G that includes all the vertices of G, and is a tree. A set of spanning trees are ***edge-disjoint***, if no two of them have any edges in common. If a given node is considered as the root of a tree, the ***depth*** of the tree is defined as the maximum distance between the root and any other node of the tree.

A communication task that will be useful in the description of our algorithms is the ***generalized multinode broadcast*** (abbreviated GMNB), where some arbitrary nodes of a network G have a total of M packets to broadcast. During a GMNB, a node may have more than one packets to broadcast (as opposed to the partial multinode broadcast task defined in [VaB94a], where each node has at most one packet to broadcast). The following lemma deals with the GMNB task in a tree.

Lemma 1: The GMNB in a tree network, under the F-D or the H-D model, requires at most $K + L - 1$ steps, where L is the diameter of the tree, and K is the total number of packets that have to be broadcast.

Proof (abbreviated): Consider the GMNB algorithm where each node transmits the packets originated at that node on all its incident links, one after the other. A node transmits each packet that it receives from its neighbors on all its incident links, except for the link over which it received it. No link remains idle if there is a packet that wants to use it, and the FIFO discipline is used (any other discipline would also work). In the case of the H-D model we also require that a link can be used only in one direction at a time. To show that the task is completed in time $K + L - 1$, note that a packet may suffer a total of at most $K - 1$ delays due to other packets. **Q.E.D.**

Given a network of diameter δ, we can always find a spanning tree $\mathcal{T}$ of depth δ rooted at some node of the network. Using the notion of the ***postorder traversal*** on $\mathcal{T}$, we define an ordering (denoted by "<") of the network nodes of $\mathcal{T}$. In the postorder traversal, the subtrees connected to the root are visited first in the order from left to right, followed by the root, with this order of traversal carried recursively on each subtree.

Given a graph G, a spanning tree $\mathcal{T}$, and a set of nodes that have a total of M packets to broadcast, we define the rank r_s of node s as $r_s = \sum_{t <_s} x_t$, where "<" is the order defined by $\mathcal{T}$, and x_t is the number of packets that node t has to broadcast. The ranks of all nodes can be computed in 2δ steps by performing a parallel prefix operation on $\mathcal{T}$. During the parallel prefix operation each node also learns the total number of packets M that have to be broadcast.

Lemma 2: Consider a graph G that has k edge-disjoint spanning trees $T_1, \dots, T_k$ with diameters $L_1 \le \cdots \le L_k = L_{\max}$, respectively. The GMNB task in G, under the F-D or the H-D model, requires at most $\frac{M}{k} + L_{\max} + 2\delta$ slots, where M is the number of packets, and δ is the diameter of G.

Proof: We first perform a parallel prefix operation to compute the rank r_s of each node s, and the total number of packets M. This requires 2δ steps. Based on the values of r_s and M, each node s can decide (in a way to be discussed shortly) the tree on which each of the packets originated at s will be broadcast. Assume that a total of n_j packets are to be broadcast along tree $T_j, j = 1, \dots, k$, where $\sum_{j=1}^{k} n_j = M$. The completion time of all transmissions on tree T_j is (by Lemma 1) at most equal to $n_j + L_j - 1$. Thus, the total time required to execute the GMNB in G satisfies

$$T_{GMNB} \le \max_{j=1,\dots,k}(n_j + L_j) - 1 + 2\delta. \tag{1}$$

One way to decide which packets are broadcast on each tree is the following. The x_s packets originated at each node s are given distinct sequense numbers in the set $\{r_s, r_s + 1, \dots, r_s + x_s - 1\}$. Node s then transmits the packet with sequense number $r_s + j - 1$, along tree $T_{(r_s+j) \bmod k}$. In this way, packets are divided equally among the trees, so that at most $\lceil M/k \rceil$ packets are broadcast on each tree. In that case, we have $n_j \le \lceil \frac{M}{k} \rceil$, which completes the proof. **Q.E.D.**

The GMNB algorithm of Lemma 2 is particularly easy to implement. The following lemma presents an algorithm that requires more computation at the nodes, but a smaller number of communication steps.

Lemma 3: Consider a graph G that has k edge-disjoint spanning trees $T_1, \ldots, T_k$ with diameters $L_1 \leq \cdots \leq L_k$, respectively. The GMNB task in G, under the F-D or the H-D model, can be executed in at most $\frac{M}{k} + \overline{L} + 2\delta$ slots, where $\overline{L} = \sum_{i=1}^{k} L_i/k$ is the average diameter of the spanning trees, M is the total number of packets that have to be broadcast, and δ is the diameter of G.

Proof: The idea is to use the algorithm of Lemma 2, but assign more packets to trees of small diameters so as to minimize the $\max_j(n_j + L_j)$ in Eq. (1). This can be achieved by running a "water-mark" algorithm locally at each node. The details are omitted. **Q.E.D.**

The algorithms of Lemmas 2 and 3 are both distributed; a node s only needs to know its rank r_s and the total number of packets M, both of which are available at s after the parallel prefix operation. For the algorithm of Lemma 3, each node also needs to know the diameters L_i, $i = 1, 2, \ldots, k$, of the trees, so that it can locally execute the watermark algorithm. Note that since all nodes run the same watermark algorithm (ties are resolved in the same way) with the same data, they all arrive at consistent results regarding the trees on which the different packets are broadcast.

3. UNIVERSAL UPPER BOUNDS ON THE THROUGHPUT AND THE DELAY

In this section, we derive upper bounds on the maximum arrival rate λ per node that can be accommodated by a given network. The bounds apply to any network topology, and they hold for any scheme that can be used to execute the broadcasts in that topology. The analysis in both the current and the next section will assume the F-D model for the network links; whenever the H-D model is used, this will be stated explicitly.

Given a general graph $G = (\mathcal{N}, \mathcal{E})$, a *partition* P of the vertices of G is a collection of non-empty disjoint subsets of $\mathcal{N}$ whose union is $\mathcal{N}$. We let $E_P(G)$ be the set of edges of G that connect nodes belonging to different constituent sets of P. We let $G_P = (P, E_P(G))$ be the graph obtained by shrinking each member of P to a single vertex, and keeping only the edges connecting nodes that belong to different sets of P (G_P may have more than one edges connecting a given pair of vertices in P). It is known ([Nas61], Theorem 1) that a graph G has k edge-disjoint spanning trees if and only if $k \leq |E_P(G)|/(|P| - 1)$ for every partition P of $\mathcal{N}$, where $|\ |$ denotes the cardinality of a set. Therefore, the maximum number of edge-disjoint spanning trees is given by

$$k_{\max} = \min_P \frac{|E_P(G)|}{|P| - 1}. \tag{2}$$

The following lemma gives an upper bound on the maximum broadcast throughput per node.

Lemma 4: A necessary condition for stability for any broadcasting scheme is $\lambda \leq \frac{2k_{\max}}{N}$.

Proof: Consider a particular partition P. Each packet generated has to undergo at least $|P| - 1$ transmissions on links of the set $E_P(G)$. Since the total rate at which broadcasts are generated in the network is equal to $N\lambda$, and there is a total of $2|E_P(G)|$ unidirectional links corresponding to the edges in $E_P(G)$, a necessary condition for stability is $\lambda N(|P| - 1) \leq 2|E_P(G)|$ for every partition P, or, using Eq. (2),

$$\lambda \leq \frac{2}{N} \min_P \frac{|E_P(G)|}{|P| - 1} = \frac{2k_{\max}}{N}. \tag{3}$$

The following lemma gives an upper bound on the maximum throughput in terms of the minimum degree $d_{\min}$ of the nodes.

Lemma 5: A necessary condition for stability for any broadcasting scheme is $\lambda \leq \frac{d_{\min}}{N-1}$.

Proof: Consider a node s of the network with indegree $d_{\min}$. Since a total of $\lambda(N-1)$ packets will have to cross the incoming links of s per unit of time, the lemma follows. **Q.E.D.**

Lemma 5 is also valid for directed (not necessarily bidirectional) graphs with $d_{\min}$ being the minimum indegree of the nodes. Equation (3) is a necessary stability condition for any network and any broadcasting scheme in that network. It should be noted, however, that there exist networks for which more stringent stability conditions hold. For example, it is easy to construct networks for which $k_{\max} = d_{\min}$. For such networks, the necessary stability condition becomes $\lambda \leq \frac{k_{\max}}{N-1}$. This implies that we cannot hope to find a broadcasting scheme that will work for *all* topologies and will be stable for $\lambda > \frac{k_{\max}}{N-1}$.

In what follows we obtain lower bounds on the average broadcast delay, which is defined as the average time that elapses between the generation of a packet at node and the time its broadcast to all the other nodes is completed. The bounds to be derived hold for any scheme that can be used to execute the broadcasts in a given network. The proofs of the lemmas use techniques similar to those developed in [StT93] for hypercube networks. Unless stated otherwise, the model assumed is the F-D model.

Lemma 6: For any graph G, the average broadcast delay $\mathcal{B}$ satisfies $\mathcal{B} = \Omega\left(\frac{1}{1-\frac{\lambda N}{2k_{\max}}}\right)$, where N is the number of nodes, λ is the rate at which broadcasts are generated at each node, and $k_{\max}$ is the maximum number of edge-disjoint spanning trees of the network.

Proof: Consider any partition $P = \{S_1, \ldots, S_{|P|}\}$ of the nodes of the graph, and let $E_P(G)$ be the set of edges connecting nodes in different sets of the partition. To obtain a lower bound on the delay, we assume that upon the generation or reception of a packet at a node in S_i, it is immediately available at all other nodes in S_i. This favorable assumption clearly underestimates the average broadcast delay. For a broadcast to be completed, each packet has to undergo at least $|P|-1$ transmissions on links of the set $E_P(G)$. Viewing the set of links $E_P(G)$ as servers, we obtain an $M/D/m$ system with arrival rate $N\lambda$ (which is the total arrival rate to the network), service time $|P|-1$, and number of servers $m = 2|E_P(G)|$. Therefore, the average broadcast delay $\mathcal{B}$ of any broadcasting scheme satisfies $\mathcal{B} \geq T_{M/D/m}$, where $T_{M/D/m}$ is the average delay of the $M/D/m$ system defined above. Using the lower bound on the delay of $M/D/m$ systems given in [Bru71] we obtain

$$\mathcal{B} = \Omega\left(\frac{1}{1-\frac{\lambda N(|P|-1)}{2|E_P(G)|}}\right). \tag{4}$$

Since Eq. (4) holds for any partition P, the lemma follows from Eq. (2). **Q.E.D.**

The following lemma gives an lower bound on the average broadcast delay in terms of the minimum degree $d_{\min}$ of the network nodes. Its proof is similar to a proof given in [StT93].

Lemma 7: Let $d_{\min}$ be the minimum degree of the nodes of a network. Then $\mathcal{B} = \Omega\left(\frac{1}{1-\frac{\lambda(N-1)}{d_{\min}}}\right)$ for any broadcasting scheme.

Lemma 7 also gives a lower bound on the average reception delay $\mathcal{R}$ for networks whose nodes have equal degrees. It also applies to directed graphs with $d_{\min}$ being the minimum of the indegrees of the nodes. The lower bound of Lemma 6 is valid for any graph, independently of the scheme used to execute the broadcasts. However, as the following corrollary shows, there exist networks for which tighter bounds on the average broadcast delay hold.

Corrollary 1: There exists a network for which the average broadcast delay is $\mathcal{B} = \Omega\left(\frac{1}{1-\frac{\lambda(N-1)}{k_{\max}}}\right)$ for any broadcasting scheme.

Proof: Consider a network that has $k_{\max} = d_{\min}$, and use Lemma 7. **Q.E.D.**

The bounds on the throughput and the average broadcast delay given in Lemmas 4 and 6 assume the F-D model. When a link can be used only in one direction at a time (H-D model), the bounds can be modified as follows.

Lemma 8 (half-duplex model): A necessary stability condition under the H-D model is $\lambda \leq k_{\max}/N$. Also the average broadcast delay $\mathcal{B}$ of any broadcasting scheme satisfies

$$\mathcal{B} = \Omega\left(\frac{1}{1-\frac{\lambda N}{k_{\max}}}\right).$$

Note that the bounds of Lemmas 5 and 7 also hold for the H-D model; since, however, we always have $k_{\max} \leq d_{\min}$, they are not as strict (for the H-D model) as those given by Lemma 8.

4. BROADCASTING ALGORITHMS

In this section we propose and analyze two dynamic broadcasting schemes for arbitrary network topologies. Both schemes use edge-disjoint spanning trees as distribution trees for the packet. In the first scheme, called the *indirect broadcasting scheme*, the broadcasts are performed by executing successive GMNB algorithms. In the second scheme, called the *direct broadcasting scheme*, each packet selects randomly one of the edge-disjoint spanning trees and is broadcast on it.

4.1. Indirect Broadcasting Scheme

One way to execute the broadcasts is by successively executing generalized multinode broadcast (GMNB) algorithms, each starting when the previous one has finished. If the graph has k edge-disjoint spanning trees with maximum diameter $L_{\max}$ and the algorithm described in Lemma 2 (where the packets are equally split among the trees) is used, the GMNB can be executed in time

$$T_{GMNB} \leq \frac{M}{k} + L_{\max} + 2\delta.$$

If the GMNB algorithm of Lemma 3 (where packets are assigned to edge-disjoint spanning trees so that the delays over all trees are nearly equal) is used, $L_{\max}$ in the preceding expression should be replaced by $\overline{L}$, where $\overline{L}$ is the average diameter of the edge-disjoint spanning trees. Note that both algorithms complete the GMNB task in at most $MX + V$ time units, where M is the number of packets and X and V are some known scalars. In the indirect broadcasting scheme, the time axis is divided into GMNB periods, each of which is in turn (conceptually) divided into two parts. The first part is called the *notification interval*, and its duration can be upper bounded by a known constant V that depends only on the size of the network and is independent of the number of packets M (in particular, if the GMNB algorithm of Lemma 2 or 3 is used, $V = L_{\max} + 2\delta$ or $V = \overline{L} + 2\delta$, respectively). During the notification interval, each active node s can be viewed as informing the other nodes that it intends to broadcast its x_s packets (this is done by merely participating in the parallel prefix operation). The second part of a GMNB period is called the *broadcast interval*, and its duration is equal to XM. Even though the duration of each GMNB period is random, it is known to all the nodes of the network, because each node learns during the broadcast interval the total number of packets M, and, from there, the duration of the following broadcast interval.

In order to analyze the performance of the indirect broadcasting scheme the following auxiliary queueing system, called *gated vacation system*, will be useful. Consider a queueing system where customers arrive at a rate of λN customers per unit of time (slot) and require $X = 1/k$ time units each in order to be served. In addition to serving customers, the server occasionally takes a vacation in order to perform some organizational work. In particular, the time axis at the server is divided into service intervals, where customers are served, and vacation intervals. When the server returns from a vacation it serves all the customers that have arrived prior to the beginning of the preceding vacation period. When all elligible customers have been served, the system takes a vacation of duration V. The gated vacation system has been analyzed in [BeG92], where the following theorem was proved.

Theorem 1: Let the arrival process of customers at the gated vacation system be a Poisson process with rate λN, and the customer service times and vacation durations be constant and equal to X and V, respectively. Then the mean waiting time in queue for this system is

$$W = \frac{\rho X}{2(1-\rho)} + \frac{V}{2} + \frac{V}{1-\rho}, \tag{5}$$

where $\rho = \lambda N X$.

The next theorem gives the average broadcast delay and the stability region of the indirect broadcasting scheme in an arbitrary network topology.

Theorem 2: Assume that for a given N-processor network there exists an algorithm that executes a GMNB in time $XM+V$, where M is the number of packets that have to be broadcast and X, V are scalars that are independent of M (they may depend on the network under consideration). Assume also that during the GMNB algorithm each node learns the value of M. Then, if the packets to be broadcast are generated at each node of the network according to a Poisson process with rate λ, independently of the other nodes, the indirect broadcasting scheme that uses this GMNB algorithm has average broadcast delay $\mathcal{B}$ that satisfies

$$\mathcal{B} \leq W + X + \min\left(W - V, \rho W\right), \tag{6}$$

where $\rho = \lambda N X$ and W is given by Eq. (5).

Proof: Each node participates in a GMNB period with all the packets generated at that node prior to the beginning of the GMNB period. The duration of the broadcast interval of a GMNB period is at most MX time units, where M is the number of elligible packets at the start of a period.

We will refer to the indirect broadcasting scheme as system "b" (for "broadcast"), and to the gated vacation system as system "a" (for "auxiliary"). Let the vacation and customer service times of system "a" be constant and equal to V and X, respectively. Consider the following analogy between systems "a" and "b". Let a service interval of system "a" correspond to a broadcast interval of system "b", and an arrival of a customer in system "a" correspond to the generation of a broadcast request in system "b". Note the similarities between the two systems. During a service interval of system "a" (or broadcast interval of system "b") all customers (or broadcast requests, respectively) can be served, provided that they arrived prior to the beginning of the current period. It is easy to see that the probability distributions of the length of the vacation intervals (which are fixed), the length of the service (or broadcast) intervals, and the number of customers (or broadcast requests) served in a service interval are identical for both systems. In particular, the duration of a vacation interval of system "a" and a notification interval of system "b" are both equal to V by construction. The duration of a broadcast interval of system "b" is equal to MX, where M is the number of packets present in the system at the beginning of the notification interval. Similarly, the duration of a service interval of system "a" is equal to MX, where M is the number of customers at the beginning of the preceding vacation interval. The only difference between the two systems is that in system "b" a broadcast is completed at the end of a GMNB period, while in system "a" customers complete service at times jX, $j = 1, 2, \ldots, M$, from the beginning of the service interval.

The waiting time W_a in queue for a packet of the auxiliary system is given by Theorem 1. Let U be the average time between the completion of service of a customer of system "a" and the end of the corresponding service interval. It can be seen that $U \leq W_a - V$. We denote by K the number of customers found in the system by an arriving customer. We then have $U \leq E(K)X = \lambda N W_a X = \rho W_a$, where we have used Little's theorem $E(K) = \lambda N W_a$. The average broadcast delay $\mathcal{B}$ satisfies

$$\mathcal{B} = W_a + X + U \leq W_a + X + \min(W_a - V, \rho W_a),$$

which completes the proof. **Q.E.D.**

The following theorem is the main result of this subsection.

Theorem 3: Let G be a (bidirectional) network that has $k_{\max}$ edge-disjoint spanning trees. Let $L_{\max}$ (or $\overline{L}$) be the maximum diameter (or the average diameter, respectively) of the edge-disjoint spanning trees, and δ be the diameter of G. The indirect broadcasting scheme that uses the GMNB algorithm of Lemma 2 is stable for

$$\lambda < \frac{k_{\max}}{N},$$

and has average broadcast delay $\mathcal{B}$ that satisfies

$$\mathcal{B} \leq \frac{\rho^2 - \rho + 2}{2k_{\max}(1-\rho)} + \frac{(L_{\max} + 2\delta)(1+\rho)(3-\rho)}{2(1-\rho)}, \tag{7}$$

where $\rho = \lambda N / k_{\max}$. Moreover, the average broadcast delay for light load satisfies

$$\mathcal{B} \leq 1.5 L_{\max} + 3\delta + \frac{1}{k_{\max}}, \qquad \rho \approx 0.$$

If the GMNB algorithm of Lemma 3 is used, then $L_{\max}$ in the above expressions should be replaced by $\overline{L}$.

Proof: If we use the GMNB algorithm described in Lemma 2 (or Lemma 3), then the average broadcast delay $\mathcal{B}$ of the indirect broadcasting scheme can be obtained from Theorem 2 by substituting $X = 1/k_{\max}$ and $V = L_{\max} + 2\delta$ (or $V = \overline{L} + 2\delta$, respectively), which after some algebraic manipulation gives Eq. (7). The indirect broadcasting scheme is, therefore, stable for $\rho = \lambda N X = \frac{\lambda N}{k_{\max}} < 1$. **Q.E.D.**

The stability region of the indirect broadcasting scheme is half of that given by the universal upper bound of Eq. (3). This is the best that we could hope for a general network, since the necessary condition of Eq. (3) is not tight for all networks. Indeed, as discussed in Section 3, a necessary stability condition for a network with $d_{\min} = k_{\max}$ is $\lambda \leq \frac{k_{\max}}{N}$ (such networks include d-dimensional meshes without wraparound and trees).

For any fixed load in the stability region, the average broadcast delay $\mathcal{B}$ is $O(L_{\max} + \delta)$ or $O(\overline{L} + \delta)$, depending on the GMNB algorithm that we use. For topologies where the edge-disjoint spanning trees can be chosen so that the maximum diameter $L_{\max}$ or the average diameter $\overline{L}$ is $O(\delta)$, the average broadcast delay of the indirect broadcasting scheme is of the optimal order (networks where this is possible include trees, and meshes of arbitrary dimension with or without wraparound). This is because, under our communication model, the network diameter is a lower bound on any broadcasting task. Since $\overline{L} \leq L_{\max}$, the indirect broadcasting scheme that uses the algorithm of Lemma 3 always performs better than the one that uses the algorithm of Lemma 2; this improvement in the performance, however, comes at the expense of a more complicated implementation and more computations at the nodes. Note also that there is a tradeoff between stability region and average broadcast delay. If we use the maximum possible number of edge-disjoint spanning trees (in order to have the best stability region), we may have to select trees of large diameter (increasing the delay for light load).

4.2. Direct Broadcasting Scheme

In this subsection, we propose and analyze an alternative dynamic broadcasting scheme, which we call the *direct broadcasting scheme*. We assume that the graph G under consideration has k edge-disjoint spanning trees, denoted by $T_j, j = 1, \ldots, k$. In the direct broadcasting scheme, each packet selects upon its generation one of the k disjoint trees, and is broadcast on it. A packet that is broadcast on tree T_j will be referred to as a packet of class j. At every slot, every node v considers each of its incident links $(v, w) \in T_j$. If v has received a packet of class j that it has neither sent already to w nor it has yet received from w, then v sends such a packet on link (v, w). If v does not have such a packet it sends nothing on (v, w). When more than one packets are elligible for transmission on link (v, w), one of them is transmitted and the remaining are queued.

The direct broadcasting scheme is a particularly simple scheme to implement. The only information carried by a packet is its class j, and each node v only has to to know which incident links are associated with each spanning tree. In what follows we evaluate the broadcast throughput and the average reception delay of the direct broadcasting scheme for an arbitrary network topology. The following lemma, proved in [StT93], will be useful in our analysis.

Lemma 9: Consider a tree $\tilde{T}$. Let s_0 be the root, $s_1, s_2, \ldots, s_n$ be the non-root nodes of $\tilde{T}$, and d_i, $i = 1, \ldots, n$, be the distance from node s_i to s_0. We assume that packets are generated at each non-root node s_i, according to a Poisson process with rate λ, and each of them is destined for node s_0. All packets require one slot for transmission over a link, and the root node s_0 can remove at most one packet per slot. The average delay D between the arrival of a packet at a node, and the time it is removed by the root is

$$D = \frac{1}{2} + \frac{\lambda n}{2(1-\lambda n)} + \frac{1}{n}\sum_{i=1}^{n} d_i. \tag{8}$$

Moreover, the system is stable if and only if $\lambda n < 1$.

Consider now the direct broadcasting scheme in a network that has k edge-disjoint spanning trees, T_j, $j = 1, 2, \ldots, k$. We assume the F-D model, where a node can transmit or receive packets over all its incident links simultaneously, and a link can be used for transmission in both directions at the same time. We let p_j be the probability with which a packet selects T_j as the spanning tree on which it will be broadcast. We also let l^j_{st} be the distance between nodes s and t using only links of tree T_j. We will initially focus on a particular node s and spanning tree T_j, and calculate the average reception delay $\mathcal{R}_j(s)$ for packets received at s over tree T_j (the average reception delay can then be calculated by averaging over all nodes s and trees T_j). In order to evaluate $\mathcal{R}_j(s)$, it is useful to view node s as the root of T_j and let $T_{j,1}(s), T_{j,2}(s), \ldots, T_{j,m}(s)$ be the subtrees in which T_j is partitioned when s is removed. We denote by $n_{j,q}(s)$ the number of nodes of tree $T_{j,q}(s)$ (therefore, $\sum_{q=1}^{m} n_{j,q}(s) = N-1$). Using Lemma 9, it can be seen that the average reception delay $\mathcal{R}_{j,q}(s)$ for packets received at node s over the subtree $T_{j,q}$ is given by

$$\mathcal{R}_{j,q}(s) = \frac{1}{2} + \frac{\lambda p_j n_{j,q}(s)}{2(1-\lambda p_j n_{j,q}(s))} + \frac{\sum_{t\in T_{j,q}} l^j_{st}}{n_{j,q}(s)}. \tag{9}$$

The average reception delay $\mathcal{R}_j(s)$ for packets received at node s over tree T_j is given by

$$\mathcal{R}_j(s) = \frac{\sum_{q=1}^{m} n_{j,q}(s)\mathcal{R}_{j,q}(s)}{N-1} = \frac{1}{2} + l^j_s + \frac{\lambda p_j}{2(N-1)}\sum_{q=1}^{m}\frac{[n_{j,q}(s)]^2}{1-\lambda p_j n_{j,q}(s)}, \tag{10}$$

where $l^j_s = \sum_{t\in T_j} l^j_{st}/(N-1)$ is the average internodal distance from every node of the network to node s using only links of T_j. The average reception delay at node s (averaged over all trees T_j, $j = 1, 2, \ldots, k$) is

$$\mathcal{R}(s) = \sum_{j=1}^{k} p_j \mathcal{R}_j(s) = \frac{1}{2} + \sum_{j=1}^{k} p_j l^j_s + \frac{\lambda}{2(N-1)}\sum_{j=1}^{k}\sum_{q=1}^{m}\frac{p_j^2[n_{j,q}(s)]^2}{1-\lambda p_j n_{j,q}(s)}. \tag{11}$$

The average reception delay $\mathcal{R}$ can be found by averaging Eq. (11) over all nodes s, that is,

$$\mathcal{R} = \frac{1}{2} + \sum_{j=1}^{k} p_j l^j + \frac{\lambda}{2N(N-1)}\sum_{s=1}^{N}\sum_{j=1}^{k}\sum_{q=1}^{m}\frac{p_j^2 n_{j,q}(s)^2}{1-\lambda p_j n_{j,q}(s)},$$

where $l^j = \sum_{s=1}^{N} l^j_s/N$ is the average internodal distance of tree T_j. The direct broadcasting scheme is stable if and only if $\lambda < (p_j n_{j,q}(s))^{-1}$, for all s, j, q. Let $\hat{s}$ be a leaf node of the spanning tree T_j. When $\hat{s}$ is viewed as the root of T_j (equivalently, when considering packets received at node $\hat{s}$

over tree T_j), there is only one subtree $T_{j,1}$ connected to $\hat{s}$, having $n_{j,1}(\hat{s}) = N-1$ nodes. Thus, the stability condition is equivalent to $\lambda < 1/(p_j(N-1))$ for all j. The maximum stability region is achieved when the number of edge-disjoint spanning trees used is equal to the maximum possible, and each packet selects with equal probability the tree on which it is broadcast (that is, when $k = k_{\max}$, and $p_j = 1/k_{\max}$ for all j). Then, the direct broadcasting scheme is stable if and only if

$$\lambda < \frac{k_{max}}{N-1}.$$

Therefore, the stability region of the direct broadcasting scheme is one half of the universal upper bound given in Eq. (3) for the F-D model, and (in view of the discussion in Section 3) it is the best that a general broadcasting scheme could achieve. The average reception delay at node s is

$$\mathcal{R}(s) = \frac{1}{2} + l_s + \frac{\lambda}{2k_{\max}(N-1)} \sum_{j=1}^{k_{\max}} \sum_{q=1}^{m} \frac{n_{j,q}(s)^2}{1 - n_{j,q}(s)\lambda/k_{\max}},$$

where $l_s = \sum_{j=1}^{k_{\max}} l_s^j / k_{\max}$. The average reception delay $\mathcal{R}$ can similarly be found as

$$\mathcal{R} = \frac{1}{2} + l + \frac{\lambda}{2k_{\max}N(N-1)} \sum_{s=1}^{N} \sum_{j=1}^{k_{\max}} \sum_{q=1}^{m} \frac{n_{j,q}(s)^2}{1 - n_{j,q}(s)\lambda/k_{\max}}, \tag{13}$$

where $l = \sum_{s=1}^{N} l_s/N$ is the mean internodal distance of the spanning trees, averaged over all nodes in a tree and over all trees. Since $n_{j,q}(s) \leq N-1$ for all j, q, and s, the average reception delay is $O(l)$ for any load in the stability region. In the usual case where the network under consideration is symmetric, Eq. (13) may be considerably simplified.

When the network is operating at a load which is considerably less than $k_{\max}/N$, maximizing the stability region may not be the primary concern. In such a case one should choose the probabilities p_j so as to minimize $\mathcal{R}$ for a given load λ. For example, when $\lambda \approx 0$, the best strategy is to broadcast all packets on the tree that has the smallest mean internodal distance.

References

[BeG92] Bertsekas, D. P., and Gallager, R. G., Data Networks, Prentice Hall, 1992.

[Bru71] Brumelle, S. L., "Some Inequalities For Parallel-Server Queues," Operations Research, vol. 19, pp. 402-413, 1971.

[Col87] Colbourn, C. J., "The Combinatorics of Network Reliability," Oxford Univ. Press, 1987.

[GrG86] Greenberg, A. G., and Goodman, J., "Sharp Approximate Models of Adaptive Routing in Mesh Networks, in Teletraffic Analysis and Computer Performance Evaluation, O. J. Boxma, J. W. Cohen, and H. C. Tijms, (Eds.), Elsevier, Amsterdam, pp. 255-270, 1986.

[GrH92] Greenberg, A. G., and Hajek, B., "Deflection Routing in Hypercube Networks," IEEE Trans. Communications, Vol. COM-35, No. 6, pp. 1070-1081, June 1992.

[Kri90] Krishna, A., Communication with Few Buffers: Analysis and Design, Ph.D. Thesis, Dept. of ECE, University of Illinois at Urbana-Champaign, December 1990.

[Nas61] Nash-Williams, C. J. A., "Edge-Disjoint Spanning Trees of Finite Graphs," Journal London Math Society, Vol. 36, 1961.

[StT93] Stamoulis, G. D., and Tsitsiklis, J. N., "Efficient Routing Schemes for Multiple Broadcasts in Hypercubes," IEEE Transactions on Parallel and Distributed Systems, Vol. 4, No. 7, July 1993.

[StT94] Stamoulis, G. D., and Tsitsiklis, J. N., "Greedy Routing in Hupercubes and Butterflies," IEEE Transactions on Communications, Vol. 44, No. 11, 1994, pp. 3051-3061.

[VaB94a] Varvarigos, E. A., and Bertsekas, D. P., "Partial Multinode Broadcast and Partial Exchange in d-Dimensional Meshes," J. of Parall. and Distrib. Comput., Vol. 23, pp. 177-189.

[VaB94b] Varvarigos, E. A., and Bertsekas, D. P., "Performance of Hypercube Routing Schemes With or Without Buffering," IEEE/ACM Trans. on Networking, Vol. 2, No. 3, pp. 299-311.

[VaB95] Varvarigos, E. A. and Bertsekas, D. P., "Dynamic Broadcasting in Parallel Computing". IEEE Trans. on Parallel and Distributed Systems, Vol. 6, No. 2, pp. 120-131, 1995.

THROUGHPUT AND DELAY FOR COPY NETWORKS WITH INTERNAL BUFFERS

P.S. Min[1], **M.V. Hegde**[2], **A. Chandra**[1], **and A. Maunder**[1]
1: Washington University in St. Louis
2: Louisiana State University

1 Introduction

The main purpose of this paper is to develop analytical models for the performance of banyan-based copy networks reported in the literature [1], wherein the copy network uses distributed buffers that are located internally to the network. In such a copy network, there is a buffer module of size b at each input of the 2×2 switching elements. A buffer module is said to be at stage i $(1 \leq i \leq n)$ if it is an input to a switching element at stage i. An example of such is the copy network used in [1] and shown in Figure 1.

At each input channel of the copy network, a fixed length packet may appear with probability p per time slot, independently of events in any other time slot, and independently of events in any other input channel. An arriving packet has associated with it the fanout request value, F, which corresponds to the number of copies to be made by the copy network. F is assumed to be an independent identically distributed (i.i.d.) random variable and $F = k$ with probability f_k ($\sum_{i=1}^{N} f_k = 1$).

A packet at stage i has associated with it a random variable $F(i)$ denoting the number of the *remaining copies* to be made before exiting from the copy network. Thus at stage 1, $F(1) = F$ where F is the fanout request value of the associated packet when it arrived. We let $f_k(i) = Prob(F(i) = k)$.

Packet replications in the network are performed on an "as needed" basis - at stage i, a packet with $F(i)$ replicates itself only if $2^{n-i} < F(i)$ since the set of destinations that can be reached by a packet in the buffer of stage $i+1$ is 2^{n-i}. If $F(i) \leq 2^{n-i}$, no replication is required at stage i. When a packet is replicated in a switching element, two copies of the packet must exit at the two outputs of the switching element in the same time slot. Packets not requiring replication are assigned at either of the two outputs in a random fashion.

A packet forwarded from stage i to stage $i+1$ is assigned $F(i+1) = F(i)$ if no replication has taken place at stage i. If the packet is replicated at stage i, one of the two copies chosen randomly is assigned $F(i+1) = \lfloor \frac{F(i)}{2} \rfloor$ and the other is assigned $F(i+1) = \lfloor \frac{F(i)+1}{2} \rfloor$. Packets associated with a point-to-point connection go through the copy network with $F(i) = 1$ for all i.

In each time slot, packets appearing at the inputs to the switching elements vie for the limited numbers of the outputs. In order for them to proceed to the next stage, they need to attain the status of *contention winners* in the manner described in what follows. Only contention winners can possibly move forward in the time slot.

Consider a switching element located at stage i. In a time slot, there may be 0, 1, or 2 packets appearing at the two inputs to this switching element. If no packet appears nothing needs to be done. When there is a packet at only one of the two inputs, this packet is a contention winner regardless of the $F(i)$ value. In this case, the packet is assigned to one output randomly if $F(i) \leq 2^{n-i}$. If $2^{n-i} < F(i)$, it is replicated and two copies of the packet are assigned to both outputs. If there are two packets appearing at the two inputs to the switching element, they are both contention winners if neither of the packets requires replication and thus can be assigned to one output each. If, on the other hand, any of the two packets appeared requires replication, then one of them is chosen randomly and is a contention winner. The status of contention winner is valid for one time slot and must be attained in each time slot.

Even if a packet at a stage is a contention winner, the movement to the next stage may be prohibited if there is no empty position in the buffer module to which the packet is routed. If the packet is to be replicated, both copies of the packet must find an empty position in their respective buffer modules to proceed. In our model, a packet is allowed to enter a full buffer module if one packet from the buffer module departs in the same time slot.

Each buffer module in the copy network can be in one of the $b+1$ possible states, $\{0, 1, 2, \ldots, b\}$. Let $b_j(i)$ denote the occupancy distribution of a buffer module at stage i, i.e., $b_j(i)$ denotes the steady state probability that there are j $(0 \le j \le b)$ packets in a buffer module at stage i at the beginning of a time slot. It can be seen that due to the symmetry in the banyan network and the i.i.d. nature of input packet arrival, the occupancy distributions for various buffer modules within the same stage are identically distributed. However, they are generally not independent. For example, given the dynamics of packet movement, if there is a long queue of packets at one of the buffer modules at the input of a switching element, then it is likely that there is a long queue at the other buffer module. However, in order to arrive at a computationally tractable model, in this paper, we assume that the occupancies of the various buffer modules in a stage are independent.

2 Simple Model

Define $\alpha(i)$ to be the probability that, at stage i, a packet chosen at random requires replication. Clearly, $\alpha(1) = \sum_{k=2^{n-1}+1}^{2^n} f_k(1)$. To find $\alpha(i)$ for $2 \le i \le n$, we determine first $f_k(i)$ such that

$$\begin{aligned}
f_k(i) &= \frac{f_k(i-1)}{\sum_{j=1}^{2^{n-i+1}} f_j(i-1) + \sum_{j=2^{n-i+1}+1}^{2^{n-i+2}} 2f_j(i-1)}, \quad k < 2^{n-i}, \\
&= \frac{f_k(i-1) + f_{2k+1}(i-1)}{\sum_{j=1}^{2^{n-i+1}} f_j(i-1) + \sum_{j=2^{n-i+1}+1}^{2^{n-i+2}} 2f_j(i-1)}, \quad k = 2^{n-i}, \\
&= \frac{f_k(i-1) + f_{2k-1}(i-1) + 2f_{2k}(i-1) + f_{2k+1}(i-1)}{\sum_{j=1}^{2^{n-i+1}} f_j(i-1) + \sum_{j=2^{n-i+1}+1}^{2^{n-i+2}} 2f_j(i-1)}, \quad 2^{n-i} < k < 2^{n-i+1}, \\
&= \frac{f_k(i-1) + f_{2k-1}(i-1) + 2f_{2k}(i-1)}{\sum_{j=1}^{2^{n-i+1}} f_j(i-1) + \sum_{j=2^{n-i+1}+1}^{2^{n-i+2}} 2f_j(i-1)}, \quad k = 2^{n-i+1}.
\end{aligned}$$

In addition, we define two probabilities concerning the movement of packets between stages: let $q(i)$ be the steady state probability that a packet is available and ready to arrive at a buffer module at stage i in a time slot (thus this packet is a contention winner at stage $i-1$), and $r(i)$ be the steady state probability that a packet from the buffer module at stage i is a contention winner and is able to move forward to stage $i+1$.

Consider a specific buffer module at stage i. Then,

$$\begin{aligned}
q(i) &= 2b_0(i-1)(1-b_0(i-1))[\frac{1}{2}(1-\alpha(i-1)) + \alpha(i-1)(1-b_b(i)(1-r(i)))] \\
&+ (1-b_0(i-1))^2[(1-\alpha(i-1))^2 + \alpha^2(i-1)(1-b_b(i)(1-r(i))) \\
&+ 2\alpha(i-1)(1-\alpha(i-1))(\frac{3}{4} - \frac{1}{2}b_b(i)(1-r(i)))] \qquad (1)
\end{aligned}$$

$$\begin{aligned} r(i) &= (1-\alpha(i))\beta(i)(1-b_b(i+1)(1-r(i+1))) + \alpha(i)\bar{\beta}(i)(1-b_b(i+1)(1-r(i+1)))^2, \\ &\qquad \text{if } 1 \le i \le n-1, \\ &= (1-\alpha(i))\beta(i) + \alpha(i)\bar{\beta}(i), \ \text{if } i = n. \end{aligned} \tag{2}$$

Here, $\beta(i)$ is the probability that a packet at stage i wins the contention given that this packet does not require replication, and $\bar{\beta}(i)$ is the probability that a packet at stage i wins the contention given that this packet requires replication. Clearly,

$$\beta(i) = b_0(i) + (1-b_0(i))(1-\alpha(i)+\frac{1}{2}\alpha(i)), \quad \bar{\beta}(i) = b_0(i) + \frac{1}{2}(1-b_0(i)). \tag{3}$$

With the state defined to be the buffer occupancy, the state transitions of a buffer module at stage i can be modeled by a birth-death process over states $\{0, 1, 2, \ldots, b\}$.

$$\begin{aligned} b_0(i) &= (1-q(i))[b_0(i) + b_1(i)r(i)], \\ b_1(i) &= q(i)[b_0(i) + b_1(i)r(i)] + (1-q(i))[b_1(i)(1-r(i)) + b_2(i)r(i)], \\ b_j(i) &= q(i)[b_{j-1}(i)(1-r(i)) + b_j(i)r(i)] \\ &\quad +(1-q(i))[b_j(i)(1-r(i)) + b_{j+1}(i)r(i)], \ \ 2 \le j \le b-1, \\ b_b(i) &= q(i)b_{b-1}(i)(1-r(i)) + b_b(i)(1-r(i)+q(i)r(i)). \end{aligned} \tag{4}$$

We determine $b_j(i)$ for $j = 0, 1, 2, \ldots b$ and $i = 1, 2, \ldots, n$ from the following iterations: With $q(1) = p$, assume initially for $i = 1, 2, \ldots, n$ that $b_0(i) = 1$ and $b_j(i) = 0$ for $j \neq 0$. $b_j(i)$, $r(i)$ and $q(i)$ can then be updated from (1) - (4). This process is repeated until $b_j(i)$'s converge.

The normalized throughput, T, of this copy network is then $T = (1-b_0(n))r(n)(1+\alpha(n))$. Delay is calculated by the use of Little's formula in the following manner: At stage i, the average number of packets in a buffer module is $\sum_{j=0}^{b} j \cdot b_j(i)$ with the average fanout request value of each packet equaling $\sum_{k=1}^{2^{n-i+1}} k \cdot f_k(i)$. Then, M, the total number of fanout request values in the entire copy network, becomes, $M = \sum_{i=1}^{n} N \cdot \left(\sum_{j=0}^{b} j \cdot b_j(i)\right)\left(\sum_{k=1}^{2^{n-i+1}} k \cdot f_k(i)\right)$. The average delay, D, from Little's formula becomes, $D = \frac{M}{NT}$.

3 Modified Model

In order to verify our model assumptions, we ran several simulations of the copy network. From the simulations, we observed that in each stage, the buffer occupancy distributions were nearly identical. However, in the buffer modules at every stage except stage 1, the arrival rate in state $b-1$ was seen to be significantly larger than the arrival rates in other states which were approximately equal. See, for example, Figure 2 for $N = 16$ and $b = 16$. Also, in the buffer modules at every stage except stage n, the departure rate from state 1 was seen significantly larger than in other states which were again approximately equal. See, for example, Figure 3 for $N = 16$ and $b = 16$.

It is easy to see that, in stage i, the increased arrival rate in state $b-1$ is a result of *pent-up* demand for a position in the stage i buffer whenever state $b-1$ is reached from state b; whenever the stage i buffer is in state b, packets from the stage $i-1$ buffers are hindered from forward progress. Such accumulation of packets in stage $i-1$ results in an increased demand for the stage i buffer whenever it becomes available. Similarly, the increased departure rate in state 1 of the stage i buffer can be seen to result from the greater likelihood, just after a sojourn of the stage i buffer in state 0, that the stage $i+1$

buffers are not full. In what follows, we modify our models to take these phenomena into account while maintaining computational ease.

We maintain the basic construct of the birth-death model in Section 2 for the evolution of the occupancy for the buffer modules. The only change from this birth-death model is the use of $\delta(i)$ as the departure rate from state 1 in stage i for $1 \leq i \leq n-1$, and $\xi(i)$ as the arrival rate to state $b-1$ in stage i for $2 \leq i \leq n$. $\xi(i)$ takes into account the increased arrival rate into state $b-1$ by considering how likely it is that state $b-1$ was reached from state b and the corresponding increase in the arrival rate due to pent up demand accumulated during the sojourn in state b. This is done as follows. For a stage i buffer, let

$$Y^l(i) = P(\text{there is a packet at a stage } i-1 \text{ buffer after } l \text{ time slots in which the stage } i \text{ buffer it was feeding was in state } b).$$

Then $Y^l(i) = \sum_{h=0}^{b} Y_h^l(i) b_h(i-1)$ where

$$Y_h^l(i) = P(\text{there is a packet at a stage } i-1 \text{ buffer after } l \text{ time slots in which the stage } i \text{ buffer it was feeding was in state } b \mid \text{this stage } i-1 \text{ buffer was in state } h \text{ at the beginning of those } l \text{ time slots}).$$

Now $Y_h^l(i)$ can be calculated from the evolution of a birth-death process describing stage $i-1$ buffer but with the departure rate being $r'(i-1)$ instead of $r(i-1)$ where $r'(i-1) = \frac{1}{2}(1-\alpha(i-1))\beta(i-1)(1-b_b(i)(1-r(i)))$ to reflect the decreased departure rate of the stage $i-1$ buffer during the sojourn of the stage i buffer in state b. If $P(i)$ denotes the transition matrix of this Markov chain, $Y_h^l(i) = 1 - P_{h,0}^l(i)$. Thus $Y^l(i) = \sum_{h=0}^{b} Y_h^l(i) b_h(i-1) = \sum_{h=0}^{b} b_h(i-1) - \sum_{h=0}^{b} P_{h,0}^l(i) b_h(i-1) = 1 - \sum_{h=0}^{b} P_{h,0}^l(i) b_h(i-1)$. Let $E^l(i)$ be defined by $E^l(i) = P$ (a packet is ready to come to the stage i buffer when the stage i buffer is in state $b-1$ after having been in state b for l previous time slots). Then

$$\begin{aligned} E^l(i) &= 2Y^l(i)(1-Y^l(i))[\frac{1}{2}(1-\alpha(i-1)) + \alpha(i-1)(1-b_b(i)(1-r(i)))] \\ &\quad +(Y^l(i))^2[(1-\alpha(i-1))^2 + \alpha^2(i-1)(1-b_b(i)(1-r(i))) \\ &\quad +2\alpha(i-1)(1-\alpha(i-1))(\frac{3}{4} - \frac{1}{2}b_b(i)(1-r(i)))]. \end{aligned}$$

Furthermore, let $A(i)$ and $G(i)$ be defined by $A(i) = P(\text{previous state was } b \mid \text{current state is } b-1)$, and $G^l(i) = P(\text{length of stay in state } b \text{ was } l)$. Then,

$$\begin{aligned} A(i) &= \frac{b_b(i)r(i)(1-\xi(i))}{b_b(i)r(i)(1-\xi(i)) + b_{b-1}(i)(\xi(i)r(i) + (1-\xi(i))(1-r(i))) + b_{b-2}(i)q(i)(1-r(i))}, \\ G^l(i) &= (1-r(i)+\xi(i)r(i))^{l-1}r(i)(1-\xi(i)). \end{aligned}$$

Thus, we can calculate $\xi(i) = A(i)\sum_{l=1}^{\infty} G^l(i)E^l(i) + (1-A(i))q(i)$. Now we address the calculation of $\delta(i)$ for $i = 1, 2, \ldots n-1$.

For a stage i buffer, let

$$X^l(i) = P(\text{stage } i+1 \text{ buffer being fed from the stage } i \text{ buffer is not in state } b \text{ after } l \text{ time slots in which this stage } i \text{ buffer was in state } 0).$$

Then $X^l(i) = \sum_{h=0}^{b} X_h^l(i) b_h(i+1)$ where

$$\begin{aligned} X_h^l(i) \quad = \quad & P(\text{stage } i+1 \text{ buffer being fed from the stage } i \text{ buffer is not in state } b \\ & \text{after } l \text{ time slots in which the stage } i \text{ buffer was in state } 0 \mid \\ & \text{stage } i+1 \text{ buffer was in state } h \text{ at the beginning of the } l \text{ time slots}). \end{aligned}$$

Now $X_h^l(i)$ can be calculated from the evolution of a birth-death process describing the stage $i+1$ buffer but with arrival rate $q'(i+1)$ instead of $q(i+1)$ where $q'(i+1) = (1-b_0(i))[\frac{1}{2}(1-\alpha(i)) + \alpha(i)(1-b_b(i+1)(1-r(i+1)))]$ to reflect the decreased arrival rate into the stage $i+1$ buffer during the sojourn of the stage i buffer in state 0. If $\bar{P}(i)$ denotes the transition matrix for this Markov chain. Then, $X_h^l(i) = 1 - \bar{P}_{h,b}^l(i)$. Thus, we have, $X^l(i) = 1 - \sum_{h=0}^{b} \bar{P}_{h,b}^l(i) b_h(i+1)$.

Let $F^l(i)$ be defined by $F^l(i) = P$ (a packet can depart from a stage i buffer when the stage i buffer is in state 1 after having been in state 0 for l previous time slots). Then $F^l(i) = (1-\alpha(i))\beta(i)X^l(i) + \alpha(i)\bar{\beta}(i)(X^l(i))^2$.

Finally, let $B(i)$ and $C^l(i)$ be defined by $B(i) = P$ (previous state of buffer was 0 | current state is 1) and $C^l(i) = P$ (length of stay at state 0 was l). Clearly

$$\begin{aligned} B(i) \quad &= \quad \frac{b_0(i)q(i)}{b_0(i)q(i) + b_1(i)(q(i)\delta(i) + (1-q(i))(1-\delta(i))) + b_2(i)(1-q(i))r(i)}, \qquad (5) \\ C^l(i) \quad &= \quad (1-q(i))^{l-1}q(i), \qquad \delta(i) = B(i)\sum_{l=1}^{\infty} C^l(i)F^l(i) + (1-B(i))r(i). \end{aligned}$$

4 Analysis of Maximum Throughput

We now determine how the maximum throughput of the copy network is limited by contention in the copying process. To remove the effect of backpressure, we let large buffers, i.e., we let $b = \infty$.

Consider, for an arbitrary switching element at stage i, one of the two buffer modules at the input. Let $\theta^1(i)$ be the probability that the considered buffer module is not empty, given that the other buffer module at the input of the same switching element is not empty. Similarly, let $\theta^0(i)$ be the probability that the considered buffer module is not empty, given that the other buffer module of the same switching element is empty. Define $\theta(i)$ to be the unconditioned probability that the considered buffer module is empty. Hence $\theta(i)$ is the result of averaging $\theta^1(i)$ and $\theta^0(i)$ with respect to the occupancy of the other buffer module.

With $b = \infty$, let $q_\infty(i)$ be the probability that there is a packet arriving to the considered buffer module in a time slot, and $r_\infty(i)$ be the probability that the head of line packet (provided that there is one such) from the buffer module can move forward in a time slot.

At stage i, the departure rate equals $r_\infty(i)\theta(i)$, which at steady state must equal $q_\infty(i)$, and thus $q_\infty(i) = r_\infty(i)\theta(i)$ where

$$r_\infty(i) = (1-\theta^1(i)) + \theta^1(i)\left[(1-\alpha(i))^2 + \alpha(i)(1-\alpha(i)) + \frac{1}{2}\alpha(i)^2\right]. \qquad (6)$$

Given the mechanism of contention, the packet dynamics, and the process of queue building, $\theta^1(i) > \theta^0(i)$ for all i. Since $\theta(i)$ is an average of these two quantities, $\theta^1(i) > \theta(i)$. Consequently $(\theta^1(i) - \theta(i))\left[1 - (1-\alpha(i))^2 - \alpha(i)(1-\alpha(i)) - \frac{1}{2}\alpha(i)^2\right] \geq 0$.

Therefore, in (6), if $\theta(i)$ is used instead of $\theta^1(i)$, an upper bound is obtained on the departure rate and thereby an upper bound on the throughput of this switching element. We note $\theta^1(i) = \theta(i)$ if the occupancies of the two buffer modules are independent. If the assumption of independent buffer occupancy is applied to every stage of the network, we obtain an upper bound on the throughput of the network. Define for each stage i,

$$\tilde{r}_\infty(i) = (1 - \theta(i)) + \theta(i)\left[(1 - \alpha(i))^2 + \alpha(i)(1 - \alpha(i)) + \frac{1}{2}\alpha(i)^2\right] \tag{7}$$

which is the departure rate of the buffer averaged over with respect to the other buffer module.

We now utilize the independence of buffer occupancy and obtain the maximum throughput and the minimum delay of the network for the corresponding rates. For stage 1, using (7) with $i = 1$ and solving for $\theta(1)$, $\theta(1) = \frac{-1\pm\sqrt{1+2p\alpha(1)(\alpha(1)-2)}}{\alpha(1)(\alpha(1)-2)}$ where we have used the fact that $p = \tilde{r}_\infty(1)\theta(1)$.

Noting that $0 \geq \alpha(1)(\alpha(1) - 2) \geq -1$ for $0 \leq \alpha(1) \leq 1$, the negative sign before the square root cannot satisfy $\theta(1) \leq 1$ and so,

$$\theta(1) = \frac{-1 + \sqrt{1 + 2q(1)\alpha(1)(\alpha(1) - 2)}}{\alpha(1)(\alpha(1) - 2)}, \quad \tilde{r}_\infty(1) = \frac{p\alpha(1)(\alpha(1) - 2)}{-1 + \sqrt{1 + 2p\alpha(1)(\alpha(1) - 2)}}.$$

The buffer module at stage 1 becomes a *Geom/Geom/*1 queueing system with the arrival rate equaling $q_\infty(1)$ and the departure rate equaling $\tilde{r}_\infty(1)$. The average time, denoted by $D_\infty(1)$, spent by a packet in the buffer module and switching element at stage 1 is,

$$D_\infty(1) = \frac{1 - q_\infty(1)}{\tilde{r}_\infty(1) - q_\infty(1)}.$$

By Burke's theorem applied to *Geom/Geom/*1 queues in tandem, the output of the first queue is a Bernoulli random process with the probability of a packet arriving at the next queue in a time slot being, $q_\infty(2) = q_\infty(1)(1-\alpha(1)) + 2q_\infty(1)\alpha(1) = q_\infty(1)(1+\alpha(1)) = p(1 + \alpha(1))$.

Thus, a buffer module at stage 2 forms another *Geom/Geom/*1 queueing system with arrival rate $q_\infty(2)$. The output of this queue is again a Bernoulli random process with the probability of a packet arriving at a stage 3 buffer module in a time slot being, $q_\infty(3) = q_\infty(2)(1 + \alpha(2)) = p(1 + \alpha(1))(1 + \alpha(2))$. In general, for $i = 2, 3, \ldots, n$, $q_\infty(i) = p\prod_{j=1}^{i}(1 + \alpha(j - 1))$), and $D_\infty(i) = \frac{1-q_\infty(i)}{\tilde{r}_\infty(i)-q_\infty(i)}$ where $D_\infty(i)$ is the delay at stage i and

$$\tilde{r}_\infty(i) = (1 - \theta(i)) + \theta(i)\left[(1 - \alpha(i))^2 + \alpha(i)(1 - \alpha(i)) + \frac{1}{2}\alpha(i)^2\right].$$

The total delay, D_∞, becomes $D_\infty = \sum_{i=1}^{n} D_\infty(i)$. As before, $\theta(i)$ can be calculated as

$$\theta(i) = \frac{-1 + \sqrt{1 + 2q_\infty(i)\alpha(i)(\alpha(i) - 2)}}{\alpha(i)(\alpha(i) - 2)}. \tag{8}$$

Now, for a stable system, the queue must become empty infinitely many times, so we must have $\theta(i) < 1$. Therefore, $\alpha(i)(\alpha(i) - 2) - \sqrt{1 + 2q_\infty(i)\alpha(i)(\alpha(i) - 2)} + 1 < 0$. By solving this as a quadratic equation with the term $\alpha(i)(\alpha(i)-2)$ as the variable, we can infer that $0 > \alpha(i)(\alpha(i)-2) > -2(1-q_\infty(i))$. The left inequality gives $0 < \alpha(i) < 2$, which is redundant except for the case of $\alpha(i) = 0$. When $\alpha(i) = 0$, no replication is made at stage i and every packet moves through without delay. Thus, stage i does not affect the maximum throughput of the copy network. The right inequality along with $\alpha(i) \leq 1$ yields, $\alpha(i) < 1 - \sqrt{2q_\infty(i) - 1}$.

Noting that $q_\infty(i) = p(1+\alpha(1))(1+\alpha(2))\ldots(1+\alpha(i-1))$,

$$\alpha(i) < 1 - \sqrt{2p(1+\alpha(1))(1+\alpha(2))\ldots(1+\alpha(i-1)) - 1}$$

and from this

$$p < \frac{(1-\alpha(i))^2 + 1}{2(1+\alpha(1))(1+\alpha(2))\ldots(1+\alpha(i-1))}.$$

Finally, the maximum throughput achievable for the copy network with $b = \infty$, which we denote as T_∞, is given by

$$T_\infty = p^*(1+\alpha(1))(1+\alpha(2))\ldots(1+\alpha(n)) \tag{9}$$

where

$$p^* = \min\left(\frac{(1-\alpha(i))^2 + 1}{2(1+\alpha(1))(1+\alpha(2))\ldots(1+\alpha(i-1))},\ \ i = 1, 2, 3, \ldots, n\right). \tag{10}$$

5 Numerical Study

Let F be a truncated geometric random variable where

$$f_k = \frac{(1-\gamma)^{k-1}\gamma}{\sum_{h=1}^{N}(1-\gamma)^{h-1}\gamma}.$$

γ is a parameter such that $0 \leq \gamma \leq 1$ and is set to result in an appropriate $E(F)$ value.

To observe the validity of the analysis given in Section 3 with respect to simulation, in Figure 4, we plot the throughput and delay obtained from simulation as well as those obtained from analysis as a function of ρ ($= E(F) \cdot p$). The figure confirm that the analysis provides reasonable approximations to the simulated values. In fact, we have observed similar accuracy in all the numerical experiments we conducted using various other values of N, b, and $E(F)$.

The models can be used to evaluate the delay and throughput of the copy network with different types of fanout distribution. For example, let F_v be the random variable corresponding for the fanout request value such that for $v = 1, 2, \ldots, \lfloor \frac{N}{2} \rfloor$

$$\begin{aligned} P(F_v = k) &= \frac{1}{2v} \quad k = 1, 2, \ldots, v \\ &= 0 \quad k = v+1, v+2, \ldots, N-v \\ &= \frac{1}{2v} \quad k = N-v+1, N-v+2, \ldots, N. \end{aligned} \tag{11}$$

For certain values of N and v, F_v represents bimodal situations where one half of the communication sessions established in the multicast switch require connections to a small number of end-points and the other half are connected to a much larger number of end-points. A result of our numerical study for F_v is shown in Figure 5.

The models in Section 3 are very accurate in the cases of most interest, i.e., where the copy network is not too congested and the delay is not too large. For large values of the delay, there is considerable dependence between the occupancies of the two buffers to the switching element particularly in the later stages. As a result there is a nonnegligible probability that one of the two buffers will be nonempty if the other one is. This increases the probability of contention and therefore reduces the departure rate. As a result, our models consistently overestimate the departure rate and therefore consistently underestimate the delay.

Next we verify the analysis of maximum throughput given in Section 4. With F set to be truncated geometric, for $N = 64$, Figure 6 shows the total delay for different values of $E(F)$. In the figure, the calculated values are compared with simulations, and we see that the analysis given in Section 4 provides a close approximation. We have observed similar accuracy in all the numerical experiments we conducted using various values of N and various fanout request distributions.

From the analysis in Section 4, the maximum throughput can be viewed as the value of ρ where the delay becomes infinite. Figure 6 confirms that the analysis provides a lower bound on the delay and an upper bound on the throughput. In addition, Figure 6 verifies that the upper bound on the maximum throughput is indeed tight.

In the figure, as ρ varies, we see that the values of $E(F)$ have little effect on D. We observed such patterns in D_∞ for all values of N. The reasons for this are as follows. The total delay is the sum of the delays in stages, each of which follow the characteristics of the delay in a *Geom/Geom/*1 queue. For the same ρ, the arrival rate and the service rate in the later stages are approximately the same. Consequently, the delays in the later stages are approximately the same. But, because in a *Geom/Geom/*1 queue the delay is not very sensitive to the arrival rate until it becomes comparable to the service rate, a significant proportion of the delay occurs in the last few stages as confirmed by Figure 7. Therefore, the impact of lower loading in the earlier stages on the delay is negligible and D_∞ is rather insensitive to changes in $E(F)$.

With $E(F) = \frac{N}{4}$, Figure 7 illustrates $D_\infty(i)$ for different values of ρ and N. It can be seen from Figure 7 that a large proportion of the delay occurs in the last few stages.

We also use our model to quantify the effects of variance in the fanout request values. Using the F_v defined in (11), the effects of different F_v's are seen in Figure 8. We see that with a constant average value, D_∞ is affected by the variance at the high ρ values only.

References

[1] Turner, J. S., "Design of a Broadcast Packet Switch Network," *IEEE Transactions on Communications*, vol. 36, no. 6, June 1988, pp. 734-743.

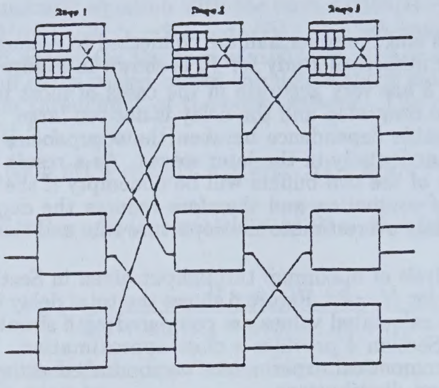

Stage 1
Stage 2
Stage 3

Figure 1. Banyan-Based Copy Network with Internal Buffers

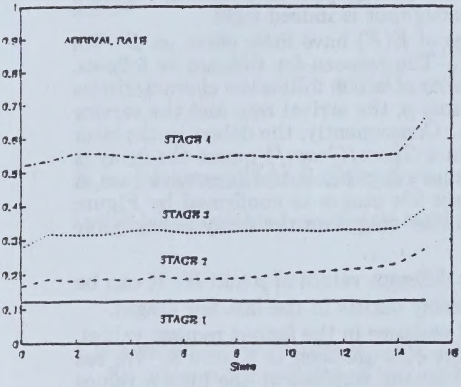


Figure 2. Observed Arrival Rate

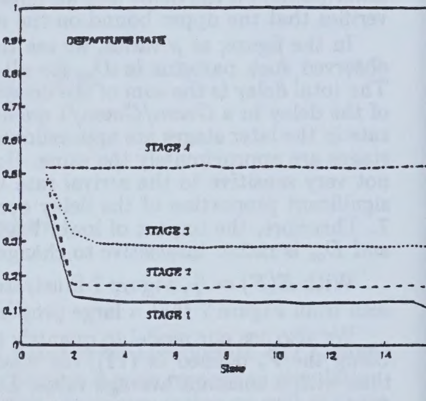


Figure 3. Observed Departure Rate

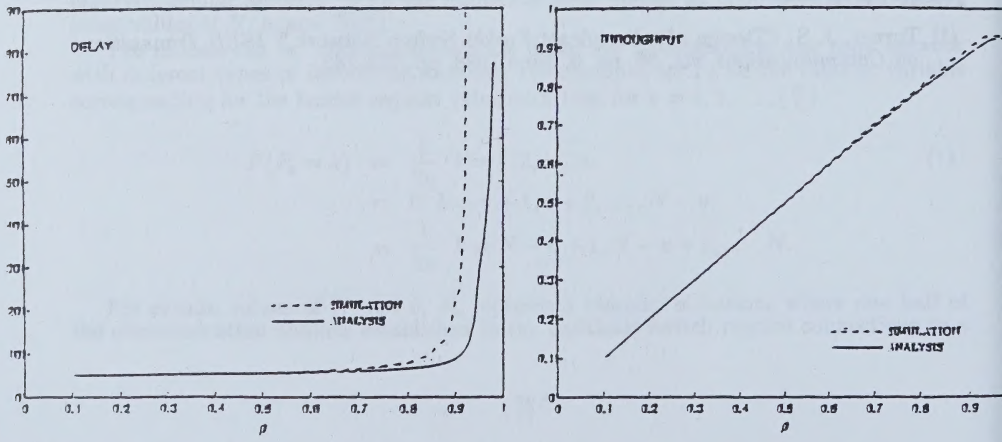


Figure 4. Truncated Geometric: $N = 32, b = 16, E(F) = 16$

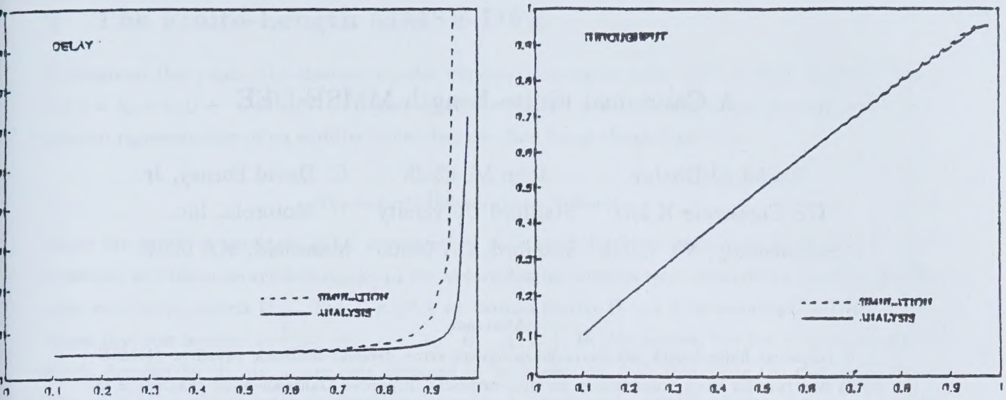
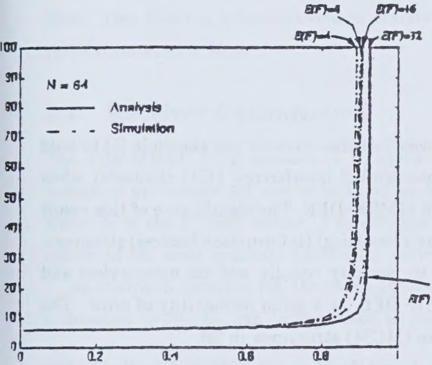


Figure 5. Bimodal: $N = 32, b = 16, n = 4$



ure 6. Delay (Function of $E(F)$, Truncated Geometric)

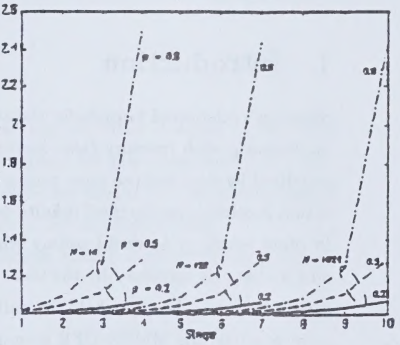


Figure 7. Delay per Stage (Truncated Geometric)

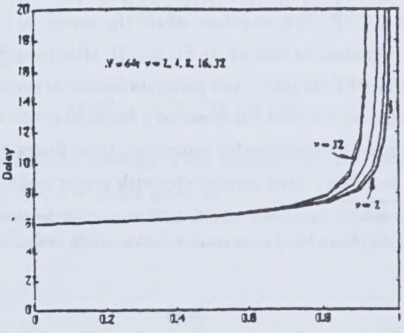


Figure 8. Delay (Function of μ, Bimodal)

A Canonical Finite-Length MMSE-DFE

Naofal Al-Dhahir
GE Corporate R &D
Schenectady, NY 12301

John M. Cioffi
Stanford University
Stanford, CA 94305

G. David Forney, Jr.
Motorola, Inc.
Mansfield, MA 02048

Abstract

A canonical finite-length minimum-mean-square-error decision-feedback equalizer (MMSE-DFE) structure for block transmission on time-invariant dispersive Gaussian-noise channels is introduced. It is shown that the optimum FIR transmit, feedforward, and feedback filters are time varying within the block. Nevertheless, we present fast algorithms, based on Cholesky factorization, to compute these FIR filters from the channel impulse response and noise correlation profile. Finally, comparisons are made with another canonical transceiver, namely vector-coding-based multichannel modulation.

1 Introduction

Shannon's celebrated formula for the capacity of a memoryless Gaussian channel was shown in [5] to hold on channels with memory (also known as dispersive or intersymbol interference (ISI) channels) when equalized by the ideal (no error propagation) infinite-length MMSE-DFE. The significance of this result is that it establishes the ideal infinite-length MMSE-DFE as a canonical (information lossless) structure. In other words, a powerful coding scheme closes the gap to capacity equally well on memoryless and on ISI channels equalized by the ideal infinite-length MMSE-DFE, at a given probability of error. The same result was shown to hold for multichannel modulation (MCM) structures in [5].

In practice, the MMSE-DFE is implemented with finite-length feedforward (FF) and feedback (FB) filters that can process only a finite block of channel output symbols and previously-detected input symbols, respectively. Unlike the infinite-length case where the error sequence (difference between the input and output of the decision device) is an *innovations* sequence [9], the finite-observation constraint causes the error sequence of the *conventional* finite-length MMSE-DFE [1] to lose its "whiteness" property [4] and the whole structure to lose its canonical property. This observation has motivated us to define an alternative finite-length MMSE-DFE structure where the canonical property is restored.

The rest of this paper is organized as follows. In Section II, after introducing the input-output model, we define a new block MMSE-DFE structure and prove its canonical property. Then, we derive the optimum settings of its FIR transmit, FF, and FB filters as a function of the channel and noise information. This is followed by presenting fast algorithms for computing these filters based on ***displacement structure theory*** [10, 15]. Section III includes a brief comparison with vector-coding-based MCM transceivers.

---

[1] By conventional we mean a symbol-by-symbol MMSE-DFE whose FIR feedforward and feedback filters are time-invariant (fixed), assuming that the channel and noise characteristics remain constant.

2 The Finite-Length MMSE-DFE

Throughout this paper, the channel impulse response is assumed to be an FIR filter of order ν; i.e., $h(D) = h_0 + h_1 D + \cdots + h_\nu D^\nu$ (ν is also called the channel memory). We adopt the baseband vector channel representation of an additive–noise discrete–time linear channel given by

$$\mathbf{y}_{k+N-1:k} = \mathbf{H}\mathbf{x}_{k+N-1:k-\nu} + \mathbf{n}_{k+N-1:k} \ , \tag{1}$$

where the output symbols $\mathbf{y}_{k+N-1:k}$ are processed on a block–by–block basis (each block consists of N symbols), and the noise symbols $\mathbf{n}_{k+N-1:k}$ are assumed to be complex zero–mean with a positive–definite noise correlation matrix $\mathbf{R}_{nn}$. The $N \times (N+\nu)$ channel matrix $\mathbf{H}$ is a *fully–windowed* Toeplitz matrix whose first row is equal to $\left[\begin{array}{ccccc} h_0 & h_1 & \cdots & h_\nu & 0_{1\times(N-1)} \end{array} \right]$. In this section, the last ν symbols in each block, denoted by $\mathbf{x}_{k-1:k-\nu}$, are only assumed to be known symbols that clear the channel memory after each block transmission to eliminate inter–block interference. Specific choices for $\mathbf{x}_{k-1:k-\nu}$ (two examples of which are given in Section III) induce other special "structure" on $\mathbf{H}$ that could be attractive from a computational point of view. The complex Gaussian zero–mean input symbols $\mathbf{x}_{k+N-1:k-\nu}$ are generated by passing a white sequence, denoted by ϵ, through an FIR (possibly time–varying) transmit filter. This filtering action introduces correlation between the input symbols characterized by the input correlation matrix $\mathbf{R}_{xx}$.

2.1 Receiver Optimization

The FIR MMSE–DFE consists of a length–N (possibly time–varying) FF filter denoted by $\mathbf{w}_i$ (that mitigates pre–cursor ISI and noise) and an FIR (possibly time–varying) FB filter denoted by $\mathbf{e}_i - \mathbf{b}_i$,[2] where $\mathbf{e}_i$ is the i^{th} unit vector. The settings of the FF and FB filters are chosen to minimize the mean square of the error sequence (difference between input and output of the decision device).

As shown in detail in [4], the mean square error of the finite–length MMSE–DFE at decision instant i, denoted by MSE_i, can be expressed as follows

$$MSE_i = \tilde{\mathbf{b}}_i^{*}(\mathbf{R}_{xx}^{-1} + \mathbf{H}^{*}\mathbf{R}_{nn}^{-1}\mathbf{H})^{-1}\tilde{\mathbf{b}}_i \overset{def}{=} \tilde{\mathbf{b}}_i^{*}\mathbf{R}_{x/y}^{\perp}\tilde{\mathbf{b}}_i \ , \tag{2}$$

where $\tilde{\mathbf{b}}_i^{*} \overset{def}{=} \left[\begin{array}{cc} 0_{1\times\Delta} & \mathbf{b}_i^{*} \end{array} \right]$, Δ $(0 \le \Delta \le N + \nu - N_b - 1)$ is the delay of the FF filter, and $\mathbf{R}_{x/y}^{\perp}$ is the covariance matrix of the error in the linear MSE estimation of $\mathbf{x}_{k+N-1:k-\nu}$ from $\mathbf{y}_{k+N-1:k}$ [9].

Now, if we define the following "lower–diagonal–upper" or *Cholesky factorization* [7]

$$[\mathbf{R}_{x/y}^{\perp}]^{-1} \overset{def}{=} \mathbf{R}_{xx}^{-1} + \mathbf{H}^{*}\mathbf{R}_{nn}^{-1}\mathbf{H} \overset{def}{=} \mathbf{L}\mathbf{D}\mathbf{L}^{*} \ , \tag{3}$$

then by combining (2) and (3), we get

$$MSE_i = \tilde{\mathbf{b}}_i^{*}\mathbf{L}^{-*}\mathbf{D}^{-1}\mathbf{L}^{-1}\tilde{\mathbf{b}}_i \ . \tag{4}$$

We claim that the optimum FB filter settings of the **canonical** finite–length MMSE–DFE for the i^{th} symbol interval (where $0 \le i \le N + \nu - 1$) are given by

---

[2] The filter $\mathbf{b}_i$ is monic and has $(N_b + 1)$ taps.

$$\tilde{\mathbf{b}}_{opt,i} = \mathbf{L}\mathbf{e}_i \ . \tag{5}$$

The resulting decision–point minimum MSE is equal to

$$MMSE_i = d_i^{-1} \ . \tag{6}$$

To prove this claim, we compute the mutual information $I(\mathbf{X};\mathbf{Y})$ [6]. Using the results of [3], we have

$$\begin{aligned} I(\mathbf{X};\mathbf{Y}) &= \log_2 |(\mathbf{R}_{xx}^{-1} + \mathbf{H}^*\mathbf{R}_{nn}^{-1}\mathbf{H})\mathbf{R}_{xx}| & (7) \\ &= \log_2 |\mathbf{L}\mathbf{D}\mathbf{L}^*\mathbf{R}_{xx}| \ : \quad \text{using (3)} & \\ &= \log_2\{|\mathbf{R}_{xx}|(\prod_{i=0}^{N+\nu-1} d_i)\} \ : \ \text{since } \mathbf{L} \text{ is a lower triangular monic matrix.} & (8) \end{aligned}$$

By expressing $|\mathbf{R}_{xx}|$ as the product of the *eigenvalues* $\sigma_{x,i}$ of $\mathbf{R}_{xx}$ we can write (8) in the following form

$$\begin{aligned} I(\mathbf{X};\mathbf{Y}) &= \log_2\{\prod_{i=0}^{N+\nu-1}(\frac{\sigma_{x,i}}{MMSE_i})\} \\ &\overset{def}{=} \log_2\{\prod_{i=0}^{N+\nu-1}(SNR_i)\} \ , \end{aligned}$$

where $SNR_i \overset{def}{=} \frac{\sigma_{x,i}}{MSE_i} = \sigma_{x,i}d_i$. Therefore, if we *define* the decision–point SNR of the finite–length MMSE–DFE to be the *geometric average* of the successive SNR_i's, i.e.,

$$SNR_{MMSE-DFE} \overset{def}{=} (\prod_{i=0}^{N+\nu-1} SNR_i)^{\frac{1}{(N+\nu)}} = (\prod_{i=0}^{N+\nu-1} (\sigma_{x,i}d_i))^{\frac{1}{(N+\nu)}} \ , \tag{9}$$

then the normalized (per input symbol) mutual information is given by

$$\overline{I}(\mathbf{X};\mathbf{Y}) \overset{def}{=} \frac{I(\mathbf{X};\mathbf{Y})}{(N+\nu)} = \log_2(SNR_{MMSE-DFE}) \ ,$$

which shows that our time–varying finite–length MMSE–DFE, depicted in Figure 1, is a canonical receiver structure.

Remarks

- The product in (9) is taken over indices i where neither $\sigma_{x,i}$ nor d_i is zero.
- When the channel and noise characteristics vary with time (intra– or inter–block variation), they need to be continuously estimated and used to compute and update the FF and FB filter settings.
- Since the FB filter is time–varying, so will the optimum FF filter be. More specifically, it can be shown that [4]

  $$\mathbf{w}_{opt,i}^* = d_i^{-1}\mathbf{e}_i^*\mathbf{L}^{-1}\mathbf{H}^*\mathbf{R}_{nn}^{-1} \quad : \quad 0 \le i \le N+\nu-1 \ , \tag{10}$$

  where the combination of the matched–filtering $(\mathbf{H}^*)$ and noise–whitening $(\mathbf{R}_{nn}^{-1})$ components can be calculated off–line and stored in memory. The tapped–delay line portion $(d_i^{-1}\mathbf{e}_i^*\mathbf{L}^{-1})$ is time–varying and of growing order. In particular, it has $(i+1)$ non–zero taps at symbol interval i.

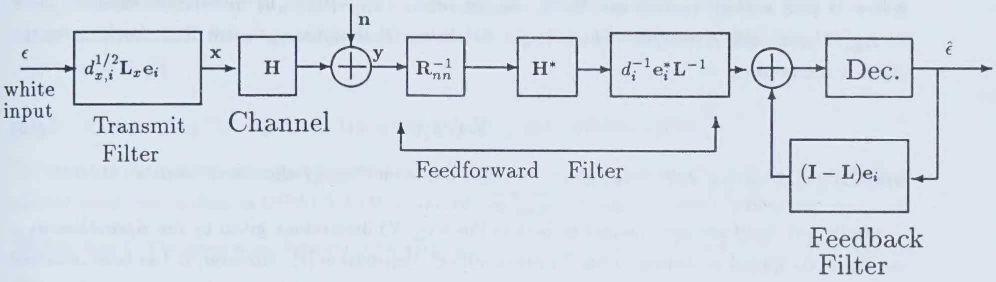


Figure 1: Block Diagram of the Canonical Block MMSE-DFE

In addition, The optimum delay, Δ_{opt}, of the FF filter is equal to i. On the other hand, the time–varying FB filter is of decreasing order; it has $(N+\nu-i)$ non–zero taps at symbol interval i.

- The action of the FB filter can be interpreted as performing a *Gram-Schmidt orthogonalizing procedure* on the linearly–independent columns of the square–root of the error covariance matrix $\mathbf{R}^{\perp}_{x/y}$. This can be seen as follows : From the Cholesky factorization of (3), we have

$$(\mathbf{R}^{\perp}_{x/y})^{-\frac{1}{2}} = \mathbf{L}\mathbf{D}^{\frac{1}{2}}\mathbf{Q}^* \quad : \text{where } \mathbf{Q} \text{ is a unitary matrix} \tag{11}$$

$$\text{Therefore,} \quad (\mathbf{R}^{\perp}_{x/y})^{\frac{1}{2}}\mathbf{L} = \mathbf{D}^{-\frac{1}{2}}\mathbf{Q}\,.$$

In words, the time–varying FB filter (given by the columns of $\mathbf{L}$) transforms the basis spanned by the columns of the square–root error covariance matrix, $(\mathbf{R}^{\perp}_{x/y})^{\frac{1}{2}}$, into an *orthogonal basis* and then scales each basis vector by $d_i^{-\frac{1}{2}} = \sqrt{MMSE_i}$. Equation (11) is widely known known as a QR-factorization [7].

- As the output block length N becomes infinite, the Cholesky factorization of (3) converges to the *spectral factorization* [5]

$$\frac{1}{R_{xx}(D)} + \frac{h(D)h^*(D^{-1})}{R_{nn}(D)} \overset{def}{=} \gamma_0^2 g(D) g^*(D^{-1})\,, \tag{12}$$

where $g(D)$ is a canonical filter (causal, monic, and minimum–phase) and $g^*(D^{-1})$ is anti–canonical. All of the time–varying forward prediction filters, $\tilde{\mathbf{b}}_{opt,i}$, converge to the linear time–invariant (LTI) *whitening filter* $g(D)$. Similarly, the time–varying tapped–delay line portion of the FF filter, $\mathbf{e}_i^*\mathbf{L}^{-1}$, converges to LTI filter $g^*(D^{-1})$, and the FF filter coefficients converge to $w^*_{opt,\infty}(D^{-1}) = \frac{h^*(D^{-1})}{\gamma_0^2 g^*(D^{-1}) R_{nn}(D)}$. Finally, all the $MMSE_i$'s converge to $\frac{1}{\gamma_0^2}$.

2.2 Transmitter Optimization

The canonical property of the MMSE–DFE receiver holds for any $\mathbf{R}_{xx}$. To optimize performance, $\mathbf{R}_{xx}$ is chosen to maximize $\bar{I}(\mathbf{X};\mathbf{Y})$ (which in turn maximizes $SNR_{MMSE-DFE}$).

If we define the eigen–decomposition

$$\mathbf{H}^*\mathbf{R}_{nn}^{-1}\mathbf{H} \overset{def}{=} \mathbf{U}\mathbf{\Sigma}\mathbf{U}^* , \tag{13}$$

where $\mathbf{U}$ is a unitary matrix and $\mathbf{\Sigma} \overset{def}{=} diag(\sigma_0, \sigma_1, \cdots, \sigma_{N-1}, 0, \cdots, 0)$ has ν zero elements since $\mathbf{H}^*\mathbf{R}_{nn}^{-1}\mathbf{H}$ is a rank–N matrix. Then, it can be shown [3] that the optimum $\mathbf{R}_{xx}$ should have the eigen–decomposition

$$\mathbf{R}_{xx} \overset{def}{=} \mathbf{U}\mathbf{\Sigma}_x\mathbf{U}^* , \tag{14}$$

where $\mathbf{\Sigma}_x \overset{def}{=} diag(\sigma_{x,0}, \sigma_{x,1}, \cdots, \sigma_{x,\bar{N}-1}, \underbrace{0, \cdots, 0}_{N+\nu-\bar{N}})$ is the input "energy allocation" matrix.

Ideally, the input energy allocated to each of the $\bar{N} (\leq N)$ dimensions, given by the eigenvalues $\sigma_{x,i}$, should be computed according to the "water–pouring" distribution [6]. However, it has been observed that on most practical channels, a *flat energy distribution* suffices, **as long as the number of usable input dimensions (or equivalently the transmission bandwidth) is optimized.** Moreover, it was shown in detail in [3] that with flat energy over the optimized bandwidth, $\mathbf{R}_{xx} \overset{def}{=} \mathbf{R}_{flat-opt}$ is given by

$$\mathbf{R}_{flat-opt} = \frac{E}{\bar{N}}\bar{\mathbf{H}}^*(\bar{\mathbf{H}}\bar{\mathbf{H}}^*)^{-1}\bar{\mathbf{H}} , \tag{15}$$

where E is the total input energy and $\bar{\mathbf{H}}$ is the $\bar{N} \times (\bar{N} + \bar{\nu})$ channel impulse response matrix corresponding to the optimum transmission bandwidth. Throughout this paper, the overbar denotes quantities sampled at the optimum symbol rate.

It was shown in [3] that $\mathbf{R}_{flat-opt}$ is a a quasi–Toeplitz matrix (corresponding to a quasi–stationary covariance process) and was synthesized using *generalized lattice filters* using the theory in [12]. Here, we present an alternative FIR realization of $\mathbf{R}_{flat-opt}$.

If we define the Cholesky factorization

$$\mathbf{R}_{flat-opt} = \mathbf{R}^{\dagger}_{flat-opt} \overset{def}{=} \mathbf{L}_x\mathbf{D}_x\mathbf{L}_x^* \overset{def}{=} \mathbf{L}_x diag(d_{x,0}, d_{x,1}, \cdots, d_{x,\bar{N}-1}, \underbrace{0, \cdots, 0}_{\bar{\nu}})\mathbf{L}_x^* ,^3 \tag{16}$$

then $\mathbf{R}_{flat-opt}$ can be synthesized exactly by a time–varying causal FIR filter whose taps at the i^{th} symbol period within a block are given by the i^{th} column of $\mathbf{L}_x$ scaled by $d_{x,i}^{\frac{1}{2}}$ for $0 \leq i \leq \bar{N} - 1$ and is all zeros for $\bar{N} \leq i \leq \bar{N} + \bar{\nu} - 1$. Since $d_{x,\bar{N}} = 0$, the synthesis procedure is exact.

2.3 Fast Computation Methods

2.3.1 Computing the FF and FB Filters of FIR MMSE-DFE

It is clear from (5) and (10) that to compute the optimum FF and FB filter settings of the time–varying canonical MMSE–DFE, we need to perform the Cholesky factorization in (3). For the cases of white (or generally stationary) noise sequence and for white, stationary, or even quasi–stationary input sequence

---

[3] $\mathbf{R}_{flat-opt} = \mathbf{R}^{\dagger}_{flat-opt}$ (the notation $(.)^{\dagger}$ denotes psuedo–inverse) since $\mathbf{R}_{flat-opt}$ is a *projection* matrix on the $\bar{N}$–dimensional row space of $\bar{\mathbf{H}}$.

(such as $\mathbf{R}_{flat-opt}$ of Section II-B), it was shown in [4] that $[\mathbf{R}^{\perp}_{x/y}]^{-1}$ is a close–to–Toeplitz matrix [10, 15] (since it involves sums, products, and inverses of Toeplitz matrices). More specifically, it can be shown that the *displacement rank* [4] of $[\mathbf{R}^{\perp}_{x/y}]^{-1}$ is not more than 4. Therefore, the Cholesky factorization of $[\mathbf{R}^{\perp}_{x/y}]^{-1}$ can be carried out in $O(4(N+\nu)^2)$ operations, as opposed to $O((N+\nu)^3)$ for a general matrix, using the *generalized Schur algorithm* described in [13, 15]. The special case of white input and noise was treated in [2].

2.3.2 Computing the Optimal Transmit Filter of FIR MMSE-DFE

To compute the optimal time–varying FIR transmit filter, we need to perform the Cholesky factorization of (16) which can be done in $O(2\bar{N}(\bar{N}+\bar{\nu}))$ computations by exploiting the special structure of $R_{flat-opt}$.

Definition 1 *The* generating function *of a semi–infinite matrix* $\mathbf{R} = [R(i,j), 0 \le i,j < \infty]$ *is defined by*

$$\begin{aligned} R(D,w) &\overset{def}{=} \begin{bmatrix} 1 & D & D^2 & \dots \end{bmatrix} \mathbf{R} \begin{bmatrix} 1 & w & w^2 & \dots \end{bmatrix}^* \\ &= \sum_{j=0}^{\infty}\sum_{i=0}^{\infty} R(i,j) D^i w^{*j} . \end{aligned}$$

It can be shown [15, 3] that the generating function of the *quasi-Toeplitz* matrix $\mathbf{R}_{flat-opt}$ has the form

$$R_{flat-opt}(D,w) = \frac{\mathbf{G}_{flat-opt}(D) \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \mathbf{G}^*_{flat-opt}(w)}{(1-dw^*)} ,$$

where $\mathbf{G}_{flat-opt}(D)$ is a row vector function with 2 elements (equal to the displacement rank of $\mathbf{R}_{flat-opt}$) and is called a **generator** [10, 13] for $\mathbf{R}_{flat-opt}$. Given $h(D)$, $\bar{N}$, and E, a computationally–efficient algorithm for computing $\mathbf{G}_{flat-opt}(D)$ was given in [3]. Starting with $\mathbf{G}_{flat-opt}(D)$, we can compute the Cholesky factorization in (16) using the *classical Schur algorithm* [10], as described next

Algorithm 1 *(classical Schur algorithm)*

Initial Condition :

$$\mathbf{G}_0(D) = \mathbf{G}_{flat-opt}(D) \overset{def}{=} \begin{bmatrix} \mathbf{u}_0(D) & \mathbf{v}_0(D) \end{bmatrix} .$$

Recursions :

For $i=0,1,\cdots,\bar{N}-1$

$$d_{x,i} = |\mathbf{G}_i(0)|^2 = |u_i(0)|^2 + |v_i(0)|^2 \quad : \textit{diagonal elements} \tag{17}$$

$$l_{x,i}(D) = D^i \mathbf{G}_i(D) \mathbf{G}_i^*(0) d_{x,i}^{-1} \quad : \textit{columns of } \mathbf{L}_x \tag{18}$$

$$k_i = \frac{\mathbf{v}_i(0)}{\mathbf{u}_i(0)} \quad : \textit{reflection coefficients} \tag{19}$$

$$D\mathbf{G}_{i+1}(D) = \mathbf{G}_i(D) \begin{bmatrix} D & -k_i \\ -k_i^* D & 1 \end{bmatrix} \frac{1}{\sqrt{1-|k_i|^2}} \quad : \textit{generator update} \tag{20}$$

---

[4] The displacement rank of a matrix $\mathbf{R}$ is equal to the rank of the matrix formed from the difference of $\mathbf{R}$ and an appropriately–defined shifted version of it [10, 15].

Output :

$$p_{opt,i}(D) = l_{x,i}(D)d_{x,i}^{-\frac{1}{2}} \qquad : \textit{FIR transmit filter}\,. \tag{21}$$

Remark :

There is a one–to–one correspondence between column vectors and polynomials in D. For example, $l_{x,i}(D) \overset{def}{=} \begin{bmatrix} 1 & D & D^2 & \cdots \end{bmatrix} \mathbf{L}_x \mathbf{e}_i$.

3 Comparison with Vector Coding

In this section, we describe another *canonical* finite–complexity transceiver structure known as ***multi-channel modulation*** (MCM) whose block diagram is shown in Figure 2. A particular form of MCM known as vector coding (VC) [11] uses *zero padding* to isolate successive input blocks. In this case, the vector representation of the input–output relationship is given by (c.f. (1))

$$\mathbf{y}_{k+N-1:k-\nu} = \mathbf{H}_{VC}\mathbf{x}_{k+N-1:k} + \mathbf{n}_{k+N-1:k-\nu}\ ,$$

where $\mathbf{H}_{VC}$ is the transpose of $\mathbf{H}$.

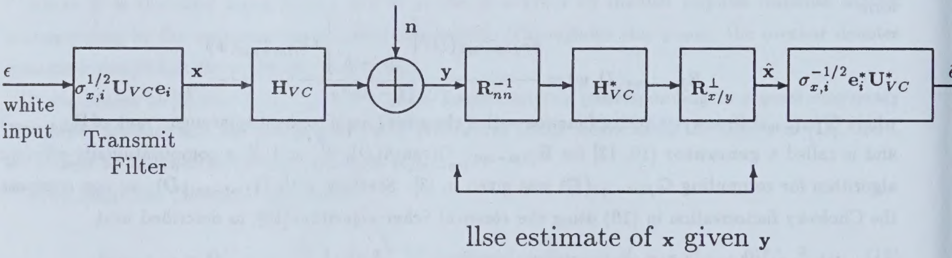

ϵ
white input
$\sigma_{x,i}^{1/2}\mathbf{U}_{VC}\mathbf{e}_i$
Transmit Filter
$\mathbf{x}$
$\mathbf{H}_{VC}$
$\mathbf{n}$
$\mathbf{y}$
$\mathbf{R}_{nn}^{-1}$
$\mathbf{H}_{VC}^*$
$\mathbf{R}_{x/y}^{\perp}$
$\hat{\mathbf{x}}$
$\sigma_{x,i}^{-1/2}\mathbf{e}_i^*\mathbf{U}_{VC}^*$
llse estimate of $\mathbf{x}$ given $\mathbf{y}$

Figure 2: Block Diagram of the VC–MCM Transceiver

MCM transceivers optimize the input correlation profile to maximize the mutual information. If we define the eigen–decomposition

$$\mathbf{H}_{VC}^*\mathbf{R}_{nn}^{-1}\mathbf{H}_{VC} \overset{def}{=} \mathbf{U}_{VC}\mathbf{\Sigma}_{VC}\mathbf{U}_{VC}^*\ , \tag{22}$$

then it can be shown that $I(\mathbf{X};\mathbf{Y})$, as given by (7), is maximized with the choice

$$\mathbf{R}_{xx,opt} = \mathbf{U}_{VC}\mathbf{\Delta}_x\mathbf{U}_{VC}^*\ , \tag{23}$$

to have the value

$$\bar{I}(\mathbf{X};\mathbf{Y}) \overset{def}{=} \frac{1}{N}I(\mathbf{X};\mathbf{Y}) \quad = \quad \frac{1}{N}\log_2 |\mathbf{I}_N + \mathbf{\Sigma}_{VC}\mathbf{\Delta}_x|$$

$$
\begin{aligned}
&= \log_2\Big(\prod_{i=1}^{N}(1+\sigma_{VC,i}\delta_{x,i})\Big)^{\frac{1}{N}} \\
&\overset{def}{=} \log_2\Big(\prod_{i=1}^{N}(1+SNR_{U,i})\Big)^{\frac{1}{N}} \\
&\overset{def}{=} \log_2(SNR_{MCM})\,, \qquad (24)
\end{aligned}
$$

where $SNR_{MCM} \overset{def}{=} (\prod_{i=1}^{N}(1+SNR_{U,i}))^{\frac{1}{N}}$. This proves the canonical property of MCM transceivers.

Remarks :

- To optimize $\bar{I}(\mathbf{X};\mathbf{Y})$ (or equivalently SNR_{MCM}), the input energy allocation matrix Σ_x must be optimized according to the "water–pouring" procedure. In this case, the product in (24) is taken over the optimum number of subchannels $\bar{N}$. As the MCM blocklength increases, this optimized $\bar{I}(\mathbf{X};\mathbf{Y})$ approaches the capacity of the underlying continuous–time channel.

- The principal computational algorithm for the finite-length block MMSE–DFE is Cholesky factorization [5] while for the MCM transceiver it is the eigen-decomposition. The advantages of Cholesky factorization are its "nesting property" [6], fast computational algorithms for "structured matrices", and its relationship to MMSE linear estimation. On the other hand, the main advantage of the eigen-decomposition, is that it is a *unitary* transformation and results in complete decoupling of the subchannels. Furthermore, in the presence of in–band interference, MCM is preferable over MMSE–DFE since it does not require a *continuous* transmission band. Since we have shown that both the block MMSE-DFE and MCM are canonical transceivers, the choice of one over the other will be determined by such implementation issues which are application–dependent.

3.1 Fast Computation Methods

3.1.1 Computing the Modulation/Demodulation Vectors of MCM

Calculating the modulating vectors for MCM transceivers requires performing the eigen–decomposition of (22) which is computationally intense since it requires $O(N^3)$ operations. For white noise, the matrix $\mathbf{H}^*_{VC}\mathbf{R}^{-1}_{nn}\mathbf{H}_{VC}$ becomes Toeplitz for which there are recursive and parallel eigen–decomposition algorithms [14, 8]. For the special case of two–tap channels, the matrix $\mathbf{H}^*_{VC}\mathbf{R}^{-1}_{nn}\mathbf{H}_{VC}$ becomes tri–diagonal and the eigenvectors become sinusoidal [1] and can be computed efficiently using the discrete sine transform.

3.1.2 Near-Optimum Modulation/Demodulation Vectors

Instead of the zero padding used in VC schemes, we can repeat the last ν information symbols at the beginning of the block. This "cyclic prefix" makes the input sequence look periodic to the channel, which gives the channel matrix a *circulant* structure. Since the eigenvectors of circulant matrices are the DFT basis vectors, the MCM modulation/ demodulation vectors can be computed efficiently using the FFT algorithm. This reduces the implementation complexity of MCM transceivers significantly. The penalty paid is a reduction in the achievable bit rate by a factor of $\frac{\nu}{(N+\nu)}$.

---

[5] For completeness, we should mention that determining the optimum transmission bandwidth of the MMSE–DFE requires computation of the non–zero eigenvalues of $\mathbf{H}^*\mathbf{R}^{-1}_{nn}\mathbf{H}$, as shown in Section II-B. For white noise, these eigenvalues become the same as those of the *Toeplitz* matrix $\mathbf{H}\mathbf{H}^*$ (c.f. Section III-A.1).

[6] The Cholesky factors of the $k \times k$ principal submatrix of an original matrix $\mathbf{R}$ are the same as the $k \times k$ submatrices of the Cholesky factors of $\mathbf{R}$.

4 Conclusions

In this paper, we extended the canonical property of the infinite–length MMSE–DFE shown in [5] to the finite–length case. The finite block of output symbols causes the optimum FIR filter settings of the MMSE–DFE to becomes time–varying. This same finite–block constraint induces a "structure" into the problem that enables an efficient computation of these FIR time–varying filters. Similarities and differences between the block MMSE–DFE and vector–coding–based MCM are delineated. By combining either one of these two canonical structures with a powerful coding scheme on an ISI Gaussian channel, the gap from capacity will be the same as that on an ISI–free Gaussian channel combined with the same code.

References

[1] N. Al-Dhahir. "An Optimized Cost-Effective Multicarrier Transceiver for Two-Ray Channels". *IEEE Signal Processing Letters*, 2:117–119, June 1995.

[2] N. Al-Dhahir and J. Cioffi. "Fast Algorithms for the Computation of the FIR MMSE-DFE". In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages IV533–IV536, San Francisco, March 1992.

[3] N. Al-Dhahir and J. Cioffi. "Block Transmission over Dispersive Channels : Transmit Filter Optimization and Realization, and MMSE-DFE Receiver Performance". to appear in IEEE Trans. on Information Theory, 1995.

[4] N. Al-Dhahir and J. Cioffi. "MMSE Decision Feedback Equalizers : Finite-Length Results". *IEEE Transactions on Information Theory*, 41(4):961–976, July 1995.

[5] J. Cioffi, G. Dudevoir, M. Eyuboglu, and G.D. Forney. "Minimum Mean-Square-Error Decision Feedback Equalization and Coding - Parts I and II". *IEEE Transactions on Communications*, October 1995.

[6] R. Gallager. *"Information Theory and Reliable Communication"*. Wiley, 1968.

[7] G. Golub and C. Van Loan. *"Matrix Computations"*. The Johns Hopkins University Press", 1989. Second Edition".

[8] Y. Hu. "Parallel Eigenvalue Decomposition for Toeplitz and Related Matrices". In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1107–1110, 1989.

[9] T. Kailath. *"Lectures on Wiener and Kalman Filtering"*. Springer-Verlag, 1981.

[10] T. Kailath. "Signal Processing Applications of Some Moment Problems". *American Mathematical Society*, 37:71–109, 1987.

[11] S. Kasturia, J. Aslanis, and J.M. Cioffi. "Vector Coding for Partial-Response Channels". *IEEE Transactions on Information Theory*, 36(4):741–762, July 1990.

[12] H. Lev-Ari. *"Nonstationary Lattice-Filter Modeling"*. PhD thesis, Stanford University, 1983.

[13] H. Lev-Ari and T. Kailath. "Triangular Factorization of Structured Hermitian Matrices". *I.Schur Methods in Operator Theory and Signal Processing*, 18:301–324, 1986.

[14] F. Noor and S. Morgera. "Recursive and Iterative Algorithms for Computing Eigenvalues of Hermitian Toeplitz matrices". *IEEE Transactions on Signal Processing*, 41(3), March 1993.

[15] A. H. Sayed. *"Displacement Structure in Signal Processing and Mathematics"*. PhD thesis, Stanford University, August 1992.

LEAST-FAVORABLE DIVERGENCE-CONSTRAINED NOISE FOR BINARY-INPUT CHANNELS *

Andrew McKellips Sergio Verdú
Department of Electrical Engineering
Princeton University
Princeton, NJ 08544, USA

Abstract

Additive noise channels with binary inputs and zero-threshold detection are considered. We study worst-case symmetric noise under the criterion of maximum error probability with constraints on both divergence and variance with respect to a given nominal noise distribution. Particular attention is focused on the cases of (a) Gaussian nominal distributions and (b) asymptotic increase in worst-case error probability when the divergence contraint tends to zero.

KEYWORDS: Detection; Hypothesis Testing; Gaussian Error Probability; Least-Favorable Noise; Kullback-Leibler Divergence.

1 Introduction

Consider a binary-input channel with additive noise N and the associated hypothesis test

$$H_0: \ Y = 1 + N \quad \text{vs.} \quad H_1: \ Y = -1 + N$$

where N is a symmetric random variable. The minimum probability of error is achieved by a zero-threshold detector and is given by

$$P = \int_1^\infty f_N(x)dx$$

where f_N denotes the probability density function of N. In many situations of interest, the distribution of N is not known exactly, and we wish to find the least-favorable noise and associated worst-case error probability over a certain class of noise distributions.

The case where the uncertainty class is the set of all noise distributions bounded in variance is particularly interesting as it quantifies the worst-case probability of error that

---

*This work was supported in part by the U.S. Army Research Office under Grant DAAH04-93-G-0219.

an optimum detector will exhibit as a function of the channel signal-to-noise ratio (*SNR*). This case was solved in [1] where it was shown that the worst-case error probability is given by

$$P(\sigma^2) = \frac{1}{2} - \frac{1}{2\sqrt{1+3\sigma_k^2}} + \frac{3(\sigma^2 - \sigma_k^2)}{2k(k+1)(2k+1)}$$

where

$$\sigma_k^2 = \frac{1}{3}(k^2-1)\ , \quad k = 1, 2, \ldots$$

and

$$\sigma_k^2 \le \sigma^2 \le \sigma_{k+1}^2$$

resulting from least-favorable noise which is realized by a mixture of two equiprobable distributions, each on a span-2 lattice.

In many cases, there is some available information about the unknown noise distribution in addition to the variance, leading to the consideration of a smaller uncertainty class. In particular, we can consider the class defined by symmetric distributions differing from the nominal by no more than a specified amount, where the discrepancy between noise distributions is quantified in a specific manner. In this paper we use the (Kullback-Leibler) divergence (see [2] for instance) as a measure of distance between the worst-case and nominal noise distributions. We will study the worst-case error probability of the zero-threshold detector achievable by any symmetric noise constrained in both divergence and variance with respect to a given nominal. It should be noted that, although the zero-threshold detector is optimum for the important special class of unimodal symmetric distributions, it does not follow that the pairing we obtain of threshold detector and least-favorable noise forms a saddle point for the game where the detector is unconstrained.

The optimization problem we face is that of finding the least-favorable noise $\hat{N}$ with density function $f_{\hat{N}}$ satisfying

$$\hat{P} = \int_1^\infty f_{\hat{N}}(x)dx = \max_{\{\text{symmetric } f_{\mathcal{N}}\}} \int_1^\infty f_{\mathcal{N}}(x)dx \tag{1}$$

subject to the constraints

$$1.\quad \int_{-\infty}^{\infty} f_{\mathcal{N}}(x)dx = 1\ , \quad \text{[true density constraint]} \tag{2}$$

$$2.\quad \int_{-\infty}^{\infty} x^2 f_{\mathcal{N}}(x)dx \le \sigma^2\ , \quad \text{[variance constraint]} \tag{3}$$

$$\text{and}\quad 3.\quad \int_{-\infty}^{\infty} f_{\mathcal{N}}(x)\log\frac{f_{\mathcal{N}}(x)}{f_N(x)}dx \le \delta \quad \text{[divergence constraint]} \tag{4}$$

where $\hat{P}$ denotes the associated worst-case probability of error.

In addition to its intrinsic interest our solution finds application in problems in data communications subject to intersymbol interference, crosstalk, multiuser interference or jamming, where the decision statistic is the sum of signal component, a noise component which is easy to characterize, and an interference component whose distribution is hard to characterize, but such that the divergence between noise plus interference and noise can be studied analytically.

The general form of least-favorable noise is presented in Section 2. A related problem of interest which is also investigated in that section arises when the noise uncertainty class is enlargened by dropping the variance constraint (3). Section 3 forms a study of the worst-case error probability when the divergence constraint tends to zero; this reflects the common situation where the noise is known to be "very close" to some specified nominal. Throughout the paper, special emphasis is placed on the important case of Gaussian nominal densities.

2 Solution Form

2.1 Divergence- and Variance-Constrained

Taking note that a symmetric threshold error probability is restricted to the interval $[0, \frac{1}{2}]$, our first step in solving (1) is to observe that, for $0 < \hat{P} < \frac{1}{2}$, equality is met in both constraints under optimality. We assume for now that $0 < \hat{P} < \frac{1}{2}$ and consider the Lagrangian

$$L(f_{\mathcal{N}}, \lambda_1, \lambda_2, \lambda_3) \;=\; \int_{-\infty}^{\infty} f_{\mathcal{N}}(x) \left\{ \frac{1}{2} 1_{\{|x| \geq 1\}} + \lambda_1 + \lambda_2 x^2 + \lambda_3 \log \frac{f_{\mathcal{N}}(x)}{f_N(x)} \right\} dx \tag{5}$$

where $1_{\{\cdot\}}$ represents the standard indicator function. The Lagrangian (5) is maximized by

$$f_{\hat{N}}(x) \;=\; f_N(x) e^{-\frac{\lambda_2 x^2}{\lambda_3}} e^{-\frac{2\lambda_1 + 1_{\{|x| \geq 1\}}}{\lambda_3}} \tag{6}$$

which is justified since $\lambda_3 > 0$ in light of the noted equality in the divergence constraint. An immediate consequence of (6) is that the optimal form obtained by ignoring the pointwise constraint $f_{\hat{N}}(x) \geq 0$ always leads to non-negative-valued realizations of $f_{\hat{N}}$, justifying the relaxing of this extraneous constraint. Further analysis also reveals that worst-case variance-constrained noise, resulting from a relaxation of the divergence constraint (4) in (1), is realized by a density function with mass at the three points $\{-1, 0, 1\}$ and is hence easy to study.

All that remains is to consider the respective special cases $\hat{P} = 0$ and $\hat{P} = \frac{1}{2}$ for which equality in the constraints has not been noted. The former case will result if and only if the nominal f_N exhibits zero error probability as a direct consequence of the divergence constraint (4), since $f_{\hat{N}}$ must remain absolutely continuous with repect to f_N in order to achieve finite divergence, and is therefore trivially uninteresting. The latter case is more involved, as there can exist an inequality in either or both constraints when $\hat{P} = \frac{1}{2}$. However, it is clear that if the worst-case noise $f_{\hat{N}}$ achieves $\hat{P} = \frac{1}{2}$, the variance of $\hat{N}$ must be no less than unity, which violates the variance constraint for nominals exhibiting $\sigma^2 < 1$. Hence, we have the useful direct observation that for any channel with $SNR > 0\ dB$, we can immediately assume equality in the constraints and resort to the optimal form (6) to solve for $f_{\hat{N}}$.

We can rewrite (6) more intuitively and with full generality as

$$f_{\hat{N}}(x) \;=\; \begin{cases} (1 - \bar{\eta}_1) f_{\bar{N}}(x)\,, & |x| < 1 \\ (1 + \bar{\eta}) f_{\bar{N}}(x)\,, & |x| \geq 1 \end{cases} \tag{7}$$

where $f_{\bar{N}}$ is a true density function expressable as

$$\begin{aligned} f_{\bar{N}}(x) &= C f_N(x) e^{-cx^2} , && (8) \\ C &= (2 \int_0^\infty f_N(x) e^{-cx^2} dx)^{-1} && (9) \end{aligned}$$

where c is a constant depending on the nominal f_N and the choice of δ.

If we denote by $\bar{P}$ the zero-threshold probability of error associated with additive noise $\bar{N}$, then the true density constraint (2) yields

$$\bar{\eta}_1 = \bar{\eta} \frac{2\bar{P}}{1 - 2\bar{P}} , \tag{10}$$

where

$$\bar{P} = \int_1^\infty C f_N(x) e^{-cx^2} dx . \tag{11}$$

The divergence constraint (4) can now be rewritten under equality in the equivalent form

$$(1 - 2\bar{P})\big(1 - \bar{\eta}\tfrac{2\bar{P}}{1-2\bar{P}}\big) \log\big(1 - \bar{\eta}\tfrac{2\bar{P}}{1-2\bar{P}}\big) + 2\bar{P}(1 + \bar{\eta}) \log(1 + \bar{\eta}) + \log C - c\sigma^2 = \delta \tag{12}$$

after some straightforward manipulation, where we have made the important observation that the term $2c \int_0^\infty x^2 f_{\hat{N}}(x) dx$ is controlled by the independent variance constraint (3) under equality, allowing us to replace it with the term $c\sigma^2$ in (12).

So, given a prespecified nominal distribution f_N and divergence constraint δ, we can determine the worst-case noise distribution $f_{\hat{N}}$ as defined by (1) by applying equations (10) and (12), where the proper value of c is determined through (3) under equality. Upon solving the above system, the worst case error probability $\hat{P}$ is given by

$$\hat{P} = \int_1^\infty f_{\hat{N}}(x) dx = (\bar{\eta} + 1)\bar{P}$$

as evidenced by (7).

2.2 Divergence-Constrained

We turn attention to the determination of worst-case noise under a solitary divergence constraint, equivalent to dropping the variance constraint from (1). That is, we wish to find $f_{\tilde{N}}$ such that

$$\tilde{P} = \int_1^\infty f_{\tilde{N}}(x) dx = \max_{\{\text{symmetric } f_{\mathcal{N}}\}} \int_1^\infty f_{\mathcal{N}}(x) dx \tag{13}$$

where $f_{\mathcal{N}}$ is subject to (2) and (4) but not (3). A useful observation is that the solution $\tilde{P}$ provides an upper bound for $\hat{P}$ in the original problem (1), and we will see in this section that solving for $\tilde{P}$ is a much easier task.

As in the analysis of Section 2.1, it is easy to show that equality is met in the divergence constraint for $0 < \tilde{P} < \frac{1}{2}$, and that the hidden pointwise constraint $f_{\tilde{N}}(x) \geq 0$ is extraneous. Subsequent Lagrangian analysis yields the optimal form

$$f_{\tilde{N}}(x) = \begin{cases} (1 - \eta \frac{2P}{1-2P}) f_N(x) , & |x| < 1 \\ (1 + \eta) f_N(x) , & |x| \geq 1 \end{cases} \tag{14}$$

which, in light of the divergence constraint (4), yields the single-variable, two-parameter equation

$$(1-2P)\left(1-\eta\frac{2P}{1-2P}\right)\log\left(1-\eta\frac{2P}{1-2P}\right)+2P(1+\eta)\log(1+\eta) \;=\; \delta_0\,, \quad (15)$$

$$\delta_0 = \min\{\delta, \log\frac{1}{2P}\}$$

to solve for η. As before, the special case $\tilde{P}=0$ results if and only if the nominal density f_N exhibits $P=0$, and is hence trivially uninteresting.

Having solved equation (15) for a given P and δ, we can write

$$\tilde{P} \;=\; (\eta+1)P \qquad (16)$$

as the solution to (13) and as an upper bound on $\hat{P}$. In terms of computational efficiency, it is beneficial that equation (15) depends on f_N only through P, limiting local dependence on the nominal density f_N. An interesting consequence is that all nominal random variables with the same threshold error probability P exhibit the same worst-case divergence-constrained probability of error $\tilde{P}$.

2.3 Gaussian Nominals

The most interesting and natural choice of nominal noise is Gaussian. In this case, we can further significantly reduce the computation required to find the worst-case noise $f_{\hat{N}}$ as outlined in Section 2.1.

Let N_σ be a zero-mean Gaussian random variable with variance σ^2. A significant savings in computation stems from the fact that the density function

$$f_{\bar{N}}(x) = C f_{N_\sigma}(x)e^{-cx^2}$$

as described by (8) is also Gaussian exhibiting variance $\bar{\sigma}^2 = \frac{\sigma^2}{1+2c\sigma^2}$. In other words, the worst-case noise $f_{\hat{N}}$ derives from first transforming the nominal Gaussian density function f_{N_σ} into a Gaussian density $f_{N_{\bar{\sigma}}}$ exhibiting lower variance, then performing the jump perturbation as described by (7), in so doing increasing the variance back to σ^2. This observation allows us to avoid the computation of C by the integral equation (9), where we can now say

$$C = \sqrt{1+2c\sigma^2}\,,$$

and allows us to express (11) as

$$\bar{P} \;=\; Q\left(\frac{\sqrt{1+2c\sigma^2}}{\sigma}\right) \qquad (17)$$

where $Q(\cdot)$ is the well-studied tail of the standard Gaussian distribution. As above, the worst-case error probability is given by

$$\hat{P} = (\bar{\eta}+1)\bar{P}\,.$$

If the variance constraint (3) is relaxed, the solution for worst-case divergence-constrained noise is simply given by (14) and (15) with $P = Q\left(\frac{1}{\sigma}\right)$.

Figures 1 and 2 depict worst-case error probability with and without a variance constraint for Gaussian nominals over a range of *SNR* values, under divergence constraints of $\delta = 0.01$ and $\delta = 0.1$ respectively. The curves corresponding to (1) under the extreme divergence-constraint values $\delta = 0$ and $\delta = +\infty$ are included for reference.

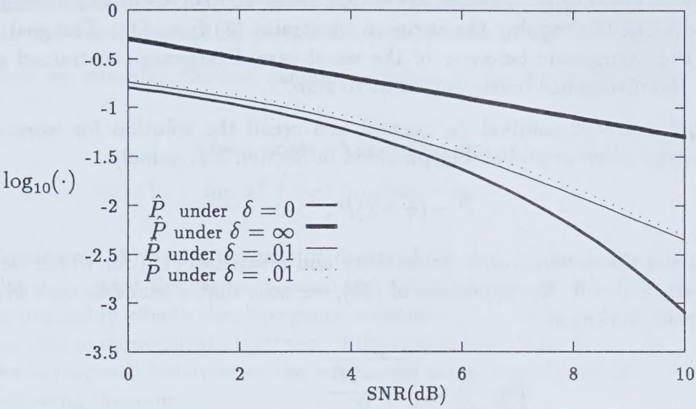


Figure 1: **Worst-case Error Probability for Gaussian Nominals, $\delta = 0.01$**

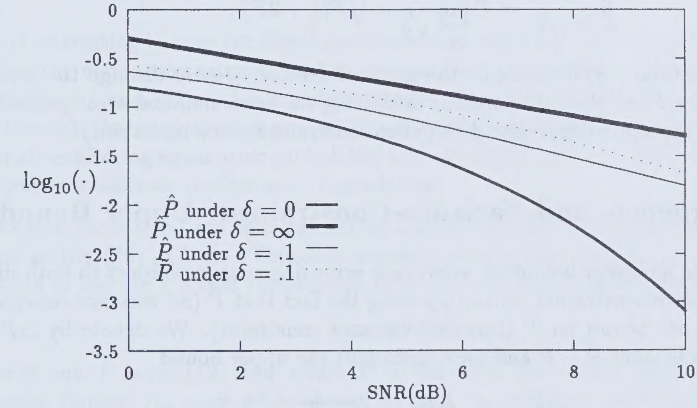


Figure 2: **Worst-case Error Probability for Gaussian Nominals, $\delta = 0.1$**

3 Asymptotic Behavior

We turn attention to the asymptotic behavior of the worst-case error performance as the divergence constraint (4) tends to zero in (1). This will convey some idea of the expected worst-case degradation in channel performance when the noise is "very close" to some nominal, for instance the usual assumption of Gaussian.

3.1 Divergence-Constrained

As before, it is interesting to consider worst-case noise subject to a lone divergence constraint, equivalent to dropping the variance constraint (3) from (1). The goal here is to quantify the asymptotic behavior of the worst-case divergence-constrained error degradation as the divergence constraint tends to zero.

We keep the choice of nominal f_N general and recall the solution for worst-case divergence-constrained error probability provided in Section 2.2, namely

$$\tilde{P} = (\eta + 1)P \ ,$$

where P represents the nominal error probability and η satisfies (15) for which we can assume $\delta < \delta_0$ since $\delta \to 0$. By inspection of (15), we note that η tends to zero with δ, allowing us to rewrite (15) as

$$\lim_{\delta \to 0} \frac{\eta}{\sqrt{\delta}} = \sqrt{\frac{1-2P}{P}} \ . \tag{18}$$

If we denote by ΔP_U the worst-case divergence-constrained error degradation $\Delta P_U = \tilde{P} - P$, (18) yields the relation

$$\lim_{\delta \to 0} \frac{\Delta P_U}{\sqrt{\delta}} = P \lim_{\delta \to 0} \frac{\eta}{\sqrt{\delta}} = \sqrt{P(1-2P)} \ . \tag{19}$$

Note that expression (19) depends on the nominal density f_N only through the nominal error probability P, so that all nominals exhibiting the same nominal error probability P also exhibit the same worst-case divergence-constrained error probability.

3.2 Divergence- and Variance-Constrained: Upper Bound

We now provide an upper bound on worst-case error degradation subject to both divergence and variance constraints, by simply using the fact that $\tilde{P}$ (no variance constraint) provides an upper bound for $\hat{P}$ (imposed variance constraint). We denote by ΔP the worst-case degradation $\hat{P} - P$ and form from (19) the upper bound

$$\lim_{\delta \to 0} \frac{\Delta P}{\sqrt{\delta}} \leq \lim_{\delta \to 0} \frac{\Delta P_U}{\sqrt{\delta}} = \sqrt{P(1-2P)} \ . \tag{20}$$

Again, the upper bound (20) depends on the nominal f_N only through the nominal error probability P. Our hope in applying the upper bound (20) is that the variance constraint has significantly less influence than the divergence constraint, a property exhibited by a class of nominals in the following section.

3.3 Divergence- and Variance-Constrained: Lower Bounds

We now turn attention to producing an asymptotic lower bound for worst-case error degradation ΔP. As in Section 3.2, we would like to limit dependence on local traits of the nominal f_N in order to obtain general results and reduce required computation; however, we take note that a constructive lower bound must meet the variance constraint (3), which is local in nature. Nonetheless, we provide two constructions which yield useful results and which depend on certain decay properties of the nominal density function.

First we consider nominal random variables N for which the following conditions hold:

$$\begin{aligned} &1. \quad \text{There exists } x^* \text{ such that } f_N(x) > 0 \text{ for all } x \geq x^* \,, && (21) \\ \text{and } &2. \quad \lim_{\bar{x}\to\infty} \bar{x}^2 \int_{\bar{x}}^{\infty} x^2 f_N(x)dx = \infty \,. && (22) \end{aligned}$$

Condition (22) limits the rate of decay of the tail of the density function f_N, and enables us to meet the variance constraint (3) by displacing area from the tail in a way which negligibly affects the divergence constraint (4). Under conditions (21) and (22), we are able to demonstrate tightness of the upper bound, providing an exact expression for the asymptotic behavior of the worst-case error degradation ΔP as summarized in the following theorem.

Theorem 3.1 *Given a nominal f_N satisfying conditions (21) and (22),*

$$\lim_{\delta\to 0} \frac{\Delta P}{\sqrt{\delta}} \;=\; \sqrt{P(1-2P)}\,. \tag{23}$$

It is interesting to note the direct match between relations (19) and (23), demonstrating that the variance constraint (3) has asymptotically negligible influence for nominals satisfying (21) and (22). Also of note is the fact that (23) depends on the nominal f_N only through the original error probability P, with the immediate consequence that all nominals exhibiting equal error probability and satisfying (21) and (22) exhibit the same asymptotic worst-case performance degradation.

We now focus attention on constructing a lower bound for Gaussian nominals, which do not satisfy (22). Given a Gaussian nominal density function f_{N_σ} with variance σ^2, we found in Section 2.3 that the solution for worst-case error probability takes the form

$$\hat{P} = (\bar{\eta}+1)\bar{P}$$

where $\bar{\eta}$ and $\bar{P}$ solve (12) and where $\bar{P}$ is the error probability associated with the Gaussian density $f_{N_{\bar{\sigma}}}$ once $\bar{\sigma}^2$ is chosen so that the variance after subjecting $f_{N_{\bar{\sigma}}}$ to transformation (7) is σ^2. We develop a lower bound P_L for $\hat{P}$ by rewriting (12) as

$$(1-2\bar{P})\left(1-\bar{\eta}\frac{2\bar{P}}{1-2\bar{P}}\right)\log\left(1-\bar{\eta}\frac{2\bar{P}}{1-2\bar{P}}\right) + 2\bar{P}(1+\bar{\eta})\log(1+\bar{\eta}) = \delta + c\sigma^2 - \frac{1}{2}\log\frac{\sigma^2}{\bar{\sigma}^2}$$

and by making the key observation that for all $c \geq 0$,

$$\begin{aligned} c\sigma^2 - \frac{1}{2}\log\frac{\sigma^2}{\bar{\sigma}^2} \;&=\; c\sigma^2 - \frac{1}{2}\log(1+2c\sigma^2) \\ &\geq\; 0\,. \end{aligned} \tag{24}$$

Hence, a valid constructive lower bound results from applying transformation (14) to a Gaussian nominal $f_{N_{\bar{\sigma}}}$ with variance $\bar{\sigma}$ such that the resulting variance is no greater than σ^2. Since the increase in variance resulting from the application of (14) to $f_{N_{\bar{\sigma}}}$ is bounded above by $2\bar{\eta}\bar{B}$, where $\bar{B} = \int_1^\infty \frac{x^2}{\sqrt{2\pi\bar{\sigma}}} e^{-\frac{x^2}{2\bar{\sigma}^2}} dx$, we satisfy the variance constraint by taking $\bar{\sigma}^2 = \sigma^2 - 2\bar{\eta}\bar{B}$. If we denote by ΔP_L the new constructive lower bound on worst-case error discrepancy, given by $\Delta P_L = \bar{\eta}\bar{P} + \bar{P} - P$, then we have the comparative asymptotic result

$$\begin{aligned}
\lim_{\delta \to 0} \frac{\Delta P_L}{\Delta P_U} &= 1 - \lim_{\delta \to 0} \frac{P - \bar{P}}{\eta P} \\
&= 1 - \lim_{\delta \to 0} \frac{Q\left(\frac{1}{\sigma}\right) - Q\left(\frac{1}{\sqrt{\sigma^2 - 2\bar{\eta}\bar{B}}}\right)}{\eta P} \\
&= 1 - \frac{Be^{-\frac{1}{2\sigma^2}}}{\sqrt{2\pi}P\sigma^3}
\end{aligned}$$

where $B = \int_1^\infty \frac{x^2}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}} dx$. This leads directly to the following theorem.

Theorem 3.2 *For Gaussian nominals,*

$$\lim_{\delta \to 0} \frac{\Delta P}{\sqrt{\delta}} \geq \sqrt{P(1 - 2P)} \left(1 - \frac{Be^{-\frac{1}{2\sigma^2}}}{\sqrt{2\pi}P\sigma^3}\right). \tag{25}$$

In Figure 3 we illustrate the use of Theorem 3.2 and result (20) by providing bounds on the quantity $\lim_{\delta \to 0} \frac{\Delta P}{\sqrt{\delta}}$ for Gaussian nominals over a range of *SNR* values.

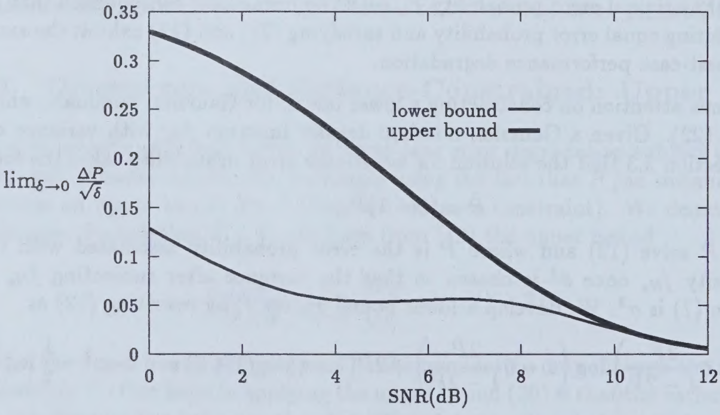


Figure 3: **Asymptotic Worst-case Error Probability for Gaussian Nominals**

4 Conclusion

We have seen that worst-case noise $f_{\hat{N}}$ under constraints of divergence and variance with respect to some nominal density f_N takes the form

$$f_{\hat{N}}(x) = \begin{cases} k_1 f_N(x) e^{-cx^2}, & |x| < 1 \\ k_2 f_N(x) e^{-cx^2}, & |x| \geq 1, \end{cases}$$

while dropping the variance constraint leads to worst-case divergence-constrained noise $f_{\tilde{N}}$ of the form

$$f_{\tilde{N}}(x) = \begin{cases} K_1 f_N(x), & |x| < 1 \\ K_2 f_N(x), & |x| \geq 1. \end{cases}$$

In Section 2 we demonstrated a procedure for computing the parameters determining $f_{\hat{N}}$ and $f_{\tilde{N}}$, including a significant reduction in computation in the case of Gaussian nominals.

Section 3 provided a study of the asymptotic behavior of the worst-case error degradation as the divergence constraint δ tends to zero. Without a variance constraint, the worst-case degradation ΔP_U was shown to satisfy the relation

$$\lim_{\delta \to 0} \frac{\Delta P_U}{\sqrt{\delta}} = \sqrt{P(1-2P)}$$

where P is the nominal probability of error. Upon imposing the variance constraint, the upper bound on worst-case degradation ΔP provided by ΔP_U was noted to be asymptotically tight for nominal densities satisfying certain decay restrictions, namely (21) and (22). For Gaussian nominals we provided a lower bound for the quantity $\lim_{\delta \to 0} \frac{\Delta P}{\sqrt{\delta}}$ which performed especially well at high values of *SNR* and demonstrated in a case of interest that ΔP vanishes proportionally to $\sqrt{\delta}$ as $\delta \to 0$.

References

[1] S.Shamai and S.Verdu, "Worst-case power-constrained noise for binary-input channels," *IEEE Transactions on Information Theory*, vol.38,no.5,pp.1494-1511, September 1992.

[2] T.M.Cover and J.A.Thomas, Elements of Information Theory, John Wiley & Sons, Inc., 1991.

Optimal and Near-Optimal Joint Channel and Data Estimation for Frequency-Selective Rayleigh Fading Channels

Yuze Zhang, Michael P. Fitz and Saul B. Gelfand
School of Electrical Engineering, Purdue University, West Lafayette, IN 47907-1285

Abstract — **This work is a systematic study of the recursive optimal and near-optimal joint channel and data estimation for stationary frequency-selective Rayleigh fading channels. An optimal joint channel and data estimator is derived using the correlation function of the channel. To reduce complexity, near-optimal algorithms are proposed. The simulation results show that the proposed algorithms outperform previously proposed algorithms.**

I. INTRODUCTION

Equalization is an important problem in many mobile wireless data communication systems, since the delay spread in the wireless channel can cause significant ISI and the Doppler spread can cause rapid time-varying fading. The delay spread can be multipath induced or introduced artificially [9, 10] in a flat fading channel. The best proposed previous demodulation schemes are the feedforward system proposed in [2], per-survivor processing (PSP) used in [3, 4] and reduced complexity MAP demodulation demonstrated in [6]. In fading channels it is often important to be able to provide soft output metrics from the demodulator since the standard architecture for mobile wireless data communication includes interleaving and coding. Rapid convergence in blind operation is also desirable in that it allows more bandwidth efficient data communication architectures. In this paper we propose algorithms that provide improved performance over these algorithms and in addition can provide soft output metrics like the BCDFE, for these algorithms all produce significantly degraded performance in important situations. The feedforward architecture of [2] uses periodically inserted groups of known symbols (often called pilot symbol assisted modulation (PSAM) for flat fading channels) to perform channel estimation and the Viterbi algorithm (VA) to make data decisions. The feedforward structure cannot be used for blind equalization, is bandwidth inefficient due to the need to transmit known symbols, and the performance exhibits a significant error floor for large Doppler spreads. PSP accomplishes demodulation with a trellis search on the possible sequences and a different channel estimator for each survivor. PSP techniques can provide blind operation but the work proposed in [3, 4] did not use model based channel estimation and hence still had significant error floors with high Doppler spreads. The model based equalizers initially suggested by Iltis in [5] and modified for symbol-by-symbol demodulation in [6] provide blind operation, soft output metrics, and better performance than PSP but still exhibit an error floor at high Doppler spreads. The error floor in these model based equalizers can be attributed to inappropriate channel models.

In the paper we address the problem of optimal demodulation on a frequency selective Rayleigh fading channel. This optimal algorithm can have a recursive but computationally complex formulation. We propose suboptimal algorithms which capture the essence of the optimal processing while resulting in more manageable complexity. The resulting algorithms have several advantages. Firstly they provide better performance than the previously proposed schemes and the error floor is greatly reduced (in some cases apparently eliminated). Secondly, they provide soft output metrics. Thirdly, the proposed algorithms do not need training sequences and are good candidates for blind equalization, although

---

[0] This work was supported by NSF Grant 9406073-NCR.

they are compatible with architectures using PSAM or training sequences in order to prevent cycle slipping or resolve rotational ambiguity. However, the pilot symbol spacing can be much smaller than in the feedforward algorithms previously mentioned. Finally, they are robust to mismatch between the Doppler spread used for channel estimation and that of the actual channel if the receiver is designed using the maximum Doppler spread. The disadvantage of the algorithms proposed here is their complexity. They either need intensive on-line computation or large memory to save parameters in order to reduce on-line computation.

In the next section, we will derive the optimal recursive joint channel and data estimation. The reduced complexity recursive algorithms are presented in Section III and IV. The results of a simulation study are presented in Section V.

II. THE OPTIMAL JOINT AND DATA ESTIMATION

Suppose the i.i.d data sequence d_k with zero mean and variance E_s takes values from a finite set Ω_d. With linear memoryless modulation and a square-root Nyquist pulse shape $u(t)$, the transmitted signal is

$$z(t) = \sum_k d_k u(t - kT) \tag{1}$$

where T is the symbol period. The autocorrelation function of $u(t)$, denoted as $R()$, satisfies $R_u(mT) = \delta(m)$. After passing through a fading channel and being corrupted by complex Additive White Gaussian Noise(AWGN) with spectral density N_0, the received signal $r(t)$ is

$$r(t) = \sum_{j=0}^{L'-1} c'_j(t) z(t - \tau_j) + v(t) \tag{2}$$

where $\tau_j = (j-1)T$, $v(t)$ is the AWGN and zero-mean complex Gaussian processes $c'_j(t), j = 1, \ldots, L'-1$, represent the stationary fading channel. The following assumptions can usually be made about the channel.

1. $R'_{c_j}(\tau) \equiv E\{c'_j(t) c'^*_j(t-\tau)\} = a'_j R_c(\tau)$, where $a'_j \geq 0, \sum_{j=0}^{L'-1} a'_j = 1$, and $R_c()$ is the autocorrelation function of a stationary process with $R_c(0) = 1$. For the well established isotropic scattering channel, we have[13] $R_c(\tau) = J_o(\omega_D \tau / T)$, where $J_o()$ is the zeroth order Bessel function of the first kind, with $\omega_D = 2\pi f_D$ and f_D being the Doppler spread of the channel normalized by the symbol period T.

2. $E\{c'_i(t) c'^*_j(t-\tau)\} = 0, i \neq j$.

3. $\int_{-\infty}^{\infty} c'_j(t) u(t - iT - \tau) u^*(t - kT) dt \approx c'_j(kT) R_u(kT - iT - \tau)$. This means that the channel varies slowly comparing with the symbol period T, which is valid for most current applications [11].

With the above assumptions, we can simply use a filter matched to $u(t)$ as front end processor and sample the signal at symbol period [11]. The sampled signal is now

$$x_k = \mathbf{d}_L^T(k) \mathbf{c}(k) + v_k \tag{3}$$

where $\mathbf{c}^T(k) \equiv [c_0(k) \ \ldots \ c_{L-1}(k)]$ with $c_j(k) = c'_j(kT)$, $\mathbf{d}_L^T(k) \equiv [d_k \ \ldots \ d_{k-L+1}]$ and v_k is complex AWGN with variance N_0,

Obviously, $\mathbf{c}(k)$ has the following property

$$E\{\mathbf{c}(k) \mathbf{c}^H(k-m)\} = R_c(mT) \mathbf{c}_{00} \tag{4}$$

where $\mathbf{c}_{00} \equiv E\{\mathbf{c}(k) \mathbf{c}^H(k)\}$ is diagonal and independent of k. This property will be useful later on when we discuss the optimal joint channel and data estimation. From now on, we will assume $\mathbf{c}_{00}$, $R_c()$, and E_s/N_o are known to the receiver. In practical systems, automatic gain control (AGC) is usually used to keep the received signal at a certain level and thus E_s/N_o can be effectively known. The isotropic model with $R_c()$ given above has long been established and has widely been used for simulation and system testing [1, 2, 8]. As for $\mathbf{c}_{00}$, it is also known when the multi-path effect is introduced artificially as in the case of transmitter diversity[9, 10]. When the multi-path effect is caused by the natural environment and thus unknown, the $\mathbf{c}_{00}$ can be estimated, which will be treated in the future work. Therefore, it is reasonable to assume the knowledge of the above parameters. Actually, this assumption has been used in the previous studies on PSAM or feedforward method [1, 2].

Let $\mathbf{d}(k) = [d_k, \ldots, d_1]^T$ be the transmitted data sequence and $\mathbf{x}(k) = [x_k, \ldots, x_1]^T$ the received signal. Suppose the decision lag is K.

Then under the minimum symbol error probability criterion, the optimal decision on d_{k-K}, given $\mathbf{x}(k)$, is

$$\hat{d}_{k-K} = \arg \max_{d_{k-K}} p(d_{k-K}|\mathbf{x}(k)) \tag{5}$$

where $p(d_{k-K}|\mathbf{x}(k))$ is the a priori probability of d_{k-K}, the soft metric needed in optimum decoding of interleaved and coded systems. It can be computed as follows,

$$\begin{aligned} p(d_{k-K}|\mathbf{x}(k)) &= \sum_{\mathbf{d}_K(k),\mathbf{d}(k-K-1)} f(\mathbf{x}(k)|\mathbf{d}(k)) \\ &\quad \cdot p(\mathbf{d}(k))/p(\mathbf{x}(k)) \end{aligned} \tag{6}$$

where $\mathbf{d}_K(k) \equiv [d_k, \ldots, d_{k-K+1}]^T$, $p(\mathbf{d}(k))$ is the a priori probability of $\mathbf{d}(k)$, $f(\mathbf{x}(k)|\mathbf{d}(k))$ is the conditional probability density function of $\mathbf{x}(k)$ given $\mathbf{d}(k)$, and $p(\mathbf{x}(k))$ is the probability of $\mathbf{x}(k)$ with

$$p(\mathbf{x}(k)) = \sum_{\mathbf{d}(k)} f(\mathbf{x}(k)|\mathbf{d}(k))p(\mathbf{d}(k))$$

Note that $f(\mathbf{x}(k)|\mathbf{d}(k))p(\mathbf{d}(k))$ can be calculated recursively by

$$\begin{aligned} f(\mathbf{x}(k)|\mathbf{d}(k))p(\mathbf{d}(k)) &= f(x_k|\mathbf{x}(k-1), \mathbf{d}(k)) \\ &\cdot p(d_k) f(\mathbf{x}(k-1)|\mathbf{d}(k-1))p(\mathbf{d}(k-1)) \end{aligned} \tag{7}$$

Since $f(x_k|\mathbf{x}(k-1), \mathbf{d}(k))$ depend on $\mathbf{x}(k-1)$ if the channel is unknown, this means that there are $|\Omega_d|^k$ possible likelihood functions $f(x_k|\mathbf{x}(k-1), \mathbf{d}(k))$. Therefore, the optimal receiver for the unknown channel is to compute the $f(\mathbf{x}(k)|\mathbf{d}(k))$ recursively for each of the $|\Omega_d|^k$ possibly transmitted data sequences $\mathbf{d}(k)$, and then compute a posteriori probability of d_{k-K} to make a decision. Since x_k and $\mathbf{x}(k-1)$ are jointly Gaussian, given $\mathbf{d}(k)$, the conditional probability density function of x_k given $\mathbf{x}(k-1)$ is also Gaussian with

$$\begin{aligned} f(x_k|\mathbf{x}(k-1), \mathbf{d}(k-1)) &= (\pi \varepsilon_{k|k-1})^{-1} \\ &\cdot \exp\left(-\frac{|x_k - \hat{x}_{k|k-1}|^2}{\varepsilon_{k|k-1}}\right) \end{aligned} \tag{8}$$

where

$$\begin{aligned} \hat{x}_{k|k-1} &\equiv E\{x_k|\mathbf{x}(k-1), \mathbf{d}(k)\} & (9) \\ \varepsilon_{k|k-1} &\equiv Var\{x_k|\mathbf{x}(k-1), \mathbf{d}(k)\} & (10) \end{aligned}$$

There are two ways to calculate $\hat{x}_{k|k-1}$ and $\varepsilon_{k|k-1}$. One is to compute them directly using a linear prediction filter (LPF) and the other is to obtain the optimal channel estimate first and and calculate the data estimate $\hat{x}_{k|k-1}$ from the channel estimate. We will discuss the former first.

For simplicity, let's first introduce the following definitions,

$$\mathbf{D}(k) \equiv diag[\mathbf{d}_L^T(k), \ldots, \mathbf{d}_L^T(1)] \tag{11}$$

$$\begin{aligned} \mathbf{c}_1^H(k) &\equiv E[\mathbf{c}(k)\mathbf{c}^H(k-1) \ \ldots \ \mathbf{c}(k)\mathbf{c}^H(1)] \\ &= [\mathbf{c}_{00}R_c(T) \ \ldots \ \mathbf{c}_{00}R_c(kT-T)] \end{aligned} \tag{12}$$

$$\begin{aligned} \mathbf{b}(k) &\equiv Cov\{\mathbf{x}(k-1), x_k|\mathbf{d}(k)\} \\ &= \mathbf{D}(k-1)\mathbf{c}_1(k)\mathbf{d}_L^T(k) & (13) \\ \mathbf{C}_x(k) &\equiv Cov\{\mathbf{x}(k)|\mathbf{D}(k)\} & (14) \\ \mathbf{a}(k) &\equiv \mathbf{C}_x^{-1}(k-1)\mathbf{b}(k) & (15) \\ e(k) &\equiv \mathbf{d}_L^T(k)\mathbf{c}_{00}\mathbf{d}_L(k) & (16) \end{aligned}$$

With the above definitions, the $\hat{x}_{k|k-1}$ and $\varepsilon_{k|k-1}$ can be expressed respectively as [15]

$$\begin{aligned} \hat{x}_{k|k-1} &= \mathbf{a}^H(k)\mathbf{x}(k-1) & (17) \\ \varepsilon_{k|k-1} &= N_o + e(k) - \mathbf{a}^H(k)\mathbf{b}(k) & (18) \end{aligned}$$

The following observations can be made about the data estimate $\hat{x}_{k|k-1}$.

1. The conditional mean $\hat{x}_{k|k-1}$ is just the linear prediction of x_k in terms of $\mathbf{x}(k-1)$, given $\mathbf{d}(k)$. And the conditional variance $\varepsilon_{k|k-1}$ is the prediction error.

2. The linear prediction filter (LPF) coefficients $\mathbf{a}(k)$ depend on $\mathbf{d}(k)$, i.e., for different survivors, the coefficients will vary. This is because $\mathbf{C}_x(k)$ and $\mathbf{b}(k)$ are functions of $\mathbf{d}(k)$.

3. The LPF changes even for a given survivor as k increases because it depends on $\mathbf{d}(k)$. Note that although the definition of $\mathbf{a}(k)$ looks like the Normal Equation of the conventional time-invariant LPF, it changes with time k. Take $\mathbf{b}(k)$ as an example. In the case of the conventional time-invariant LPF, we have $\mathbf{b}^H(k) = [b_k^* \ \mathbf{b}^H(k-1)]$. This is no longer the case for the time-varying LPF and there is no such simple relationship between $\mathbf{b}(k)$ and $\mathbf{b}(k-1)$.

All the parameters of the LPF can be updated recursively, i.e., $\mathbf{C}_x^{-1}(k)$, $\varepsilon_{k+1|k}$ and $\mathbf{a}(k+1)$ can be calculated in terms of $\mathbf{C}_x^{-1}(k-1)$, $\mathbf{a}(k-1)$ and $\varepsilon_{k|k-1}$, except $\mathbf{b}(k+1)$ and $e(k)$, which have to be calculated each time a signal is received. Actually, all the recursive relationships have been given above except that for $\mathbf{C}_x^{-1}(k)$, which is given by

$$\mathbf{C}_x^{-1}(k) = \varepsilon_{k|k-1}^{-1}\begin{bmatrix} 1 & -\mathbf{a}^H(k) \\ -\mathbf{a}(k) & \mathbf{a}(k)\mathbf{a}^H(k) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{C}_x^{-1}(k-1) \end{bmatrix} \quad (19)$$

Note that $\mathbf{C}_x^{-1}(k)$ contains all the parameters needed for the LPF at time k, namely, $\mathbf{a}(k), \varepsilon_{k|k-1}$, and $\mathbf{C}_x^{-1}(k-1)$, and thus all the information needed for LPF updating, except for $e(k+1)$ and $\mathbf{b}(k+1)$.

Now let's look at the other way to compute the data estimate. Traditionally, people used to obtain the channel estimate first and then calculate the data estimate from it. For the optimal receiver, the optimal channel estimate at time k for the survivor $\mathbf{d}(k-1)$ is given as

$$\hat{\mathbf{c}}(k|k-1) \equiv E\{\mathbf{c}(k)|\mathbf{x}(k-1), \mathbf{d}(k-1)\} \quad (20)$$

and the covariance matrix of the channel estimate

$$\mathbf{Q}(k|k-1) \equiv Cov\{\mathbf{c}(k)|\mathbf{x}(k-1), \mathbf{d}(k-1)\} \quad (21)$$

The data estimate, or the conditional mean of the likelihood function is given by

$$\hat{x}_{k|k-1} = \mathbf{d}_L^T(k)\hat{\mathbf{c}}(k|k-1) \quad (22)$$

and the conditional variance

$$\varepsilon_{k|k-1} = N_o + \mathbf{d}_L^H(k)\mathbf{Q}(k|k-1)\mathbf{d}_L^*(k) \quad (23)$$

Using the results as shown in Equation (17), (15), (13) and (18), we can obtain the channel estimate

$$\hat{\mathbf{c}}(k|k-1) = \mathbf{g}_1^H(k)\mathbf{x}(k-1) \quad (24)$$

and its covariance matrix

$$\mathbf{Q}(k|k-1) = \mathbf{c}_{00} - \mathbf{g}_1^H(k)\mathbf{D}(k)\mathbf{c}_1(k) \quad (25)$$

where

$$\mathbf{g}_1(k) \equiv \mathbf{c}_1^H(k)\mathbf{D}^H(k)\mathbf{C}_x^{-1}(k-1) \quad (26)$$

The following comments can be made about the optimal channel estimate. First, due to the Gaussian property of the channel, $\hat{\mathbf{c}}(k|k-1)$ is a linear estimate and the $\mathbf{g}_1(k)$ is the so-called Wiener filter's coefficients. Second, just as the LPF, the Wiener filter (WF)is different for different data sequences due to its dependency on the data and it changes as k grows. Finally, the WF and the LPF are closely related and just the different aspects of the same thing. The relationship between their coefficients is given by $\mathbf{a}(k) = \mathbf{g}_1(k)\mathbf{d}_L^*(k)$. With this relationship, the WF can also be updated using the recursive algorithm for the LPF.

Apparently, the complexity of the optimal joint data and channel estimation is very high. On the one hand, the length of the LPF or WF grows with the number of received symbols. On the other hand, the number of possibly transmitted data sequences increases exponentially with the number of received symbols. In the following sections, we will first discuss the algorithms to reduce the complexity for each data sequence and then discuss the techniques to reduce the number of hypothetically transmitted data sequences, or survivors.

III. COMPLEXITY REDUCTION FOR EACH SURVIVOR

Since there are two ways to calculate likelihood, we can simplify its complexity with two approaches accordingly. One is to use a fixed-length LPF for data estimation, and the other is to use an IIR filter for channel estimation. We will discuss the former first.

The length of the optimal LPF increases with time because it uses all the received signals. Thus an obvious way to reduce the complexity is to fix the length of the LPF and uses only the most recently received signals. More specifically, we can estimate x_k from $\mathbf{x}_N(k-1) \equiv [x_{k-1}, \ldots, x_{k-N}]^T$ by

$$\hat{x}_{k|k-1}(N) \equiv E\{x_k|\mathbf{x}_N(k-1), \mathbf{D}_{N+1}(k),\} \quad (27)$$

where

$$\mathbf{D}_{N+1}(k) \equiv diag\{\mathbf{d}_L^T(k), \ldots, \mathbf{d}_L^T(k-N)\} \quad (28)$$

Defining a few quantities similar to those used for the optimal LPF

$$\mathbf{C}_x(k-1, N) \equiv Cov\{\mathbf{x}_N(k-1)|\mathbf{D}_N(k-1)\} \quad (29)$$

$$\begin{aligned}\mathbf{b}_N^H(k) &\equiv Cov\{x_k, \mathbf{x}_N(k-1)|\mathbf{D}_{N+1}(k)\} \\ &= \mathbf{d}_L^T(k)\mathbf{c}_1^H(N)\mathbf{D}_N^H(k-1) \qquad (30)\\ \mathbf{a}_N(k) &\equiv \mathbf{C}_x(k-1,N)\mathbf{b}_N(k) \qquad (31)\end{aligned}$$

we can obtain the data estimate, or the output of the LPF with the fixed-length N as follows [15],

$$\hat{x}_{k|k-1}(N) = \mathbf{a}_N^H(k)\mathbf{x}_N(k) \qquad (32)$$

And similarly the linear prediction error can be obtained as

$$\begin{aligned}\varepsilon_{k|k-1}(N) &\equiv Var\{x_k|\mathbf{x}_N(k-1), \mathbf{D}_{N+1}(k)\} \\ &= N_o + e(k) - \mathbf{a}_N^H(k)\mathbf{b}_N(k) \qquad (33)\end{aligned}$$

With these results, we can replace $\hat{x}_{k|k-1}$ and $\varepsilon_{k|k-1}$ with $\hat{x}_{k|k-1}(N)$ and $\varepsilon_{k|k-1}(N)$ and then calculate the conditional probability density functions $f(x_k|\mathbf{x}_N(k-1), \mathbf{d}_N(k))$ to replace $f(x_k|\mathbf{x}(k-1), \mathbf{d}(k))$.

Several observations can be made about the LPF with fixed-length N.

1. The fixed-length LPF is the same with the optimal LPF if $R_c(mT) = 0$ for $m > N$.
2. For a given time k, the prediction error $\varepsilon_{k|k-1}(N)$ decreases with the length of the LPF N [15].
3. Since the LPF's coefficients are independent of the received signals, they can be computed in advance and saved. This will reduce the complexity of on-line computation to the order $O(N)$. However, this means saving $|\Omega_d|^N$ filters.

The LPF with fixed length can also be updated recursively, whose complexity has the order of $O(N^2)$. Similar to the optimal LPF, the main cost lies in the computation of $\mathbf{C}_x^{-1}(k,N)$ from $\mathbf{C}_x^{-1}(k-1,N)$. Actually it takes two steps to complete the conversion: first from $\mathbf{C}_x^{-1}(k-1,N)$ to $\mathbf{C}_x^{-1}(k-1,N-1)$ and then from $\mathbf{C}_x^{-1}(k-1,N-1)$ to $\mathbf{C}_x^{-1}(k,N)$. In comparison, for the optimal LPF we can come to $\mathbf{C}_x^{-1}(k)$ from $\mathbf{C}_x^{-1}(k-1)$ in one step. Because of the length of the paper, we can not describe the algorithm here, which can be found in [15].

The LPF of fixed length discussed above is an FIR filter. The more observations used, the better the performance, but it is also more complicated. Therefore, we consider a different approach now. We will show in the following that the optimal channel estimator becomes an IIR type filter under certain conditions, which reduces the complexity to the order of $O(L^2)$ for each data sequence.

It turns out that the channel estimate $\hat{\mathbf{c}}(k+1|k)$ and its covariance matrix $\mathbf{Q}(k+1|k)$ can be obtained respectively as [15]

$$\begin{aligned}\hat{\mathbf{c}}(k+1|k) = \mathbf{g}_2^H(k)\mathbf{x}(k-1) + \varepsilon_{k|k-1}^{-1}(x_k - \hat{x}_{k|k-1}) \\ \cdot [\mathbf{c}_{01}^H - \mathbf{g}_2^H(k)\mathbf{D}(k-1)\mathbf{c}_1(k)]\mathbf{d}_L^*(k)(\end{aligned}$$

and

$$\begin{aligned}\mathbf{Q}(k+1|k) = \mathbf{c}_{00} - \mathbf{g}_2^H(k+1)\mathbf{D}(k-1)\mathbf{c}_2(k) \\ -\varepsilon_{k|k-1}^{-1}[\mathbf{c}_{01}^H - \mathbf{g}_2^H(k)\mathbf{D}(k-1)\mathbf{c}_1(k)] \\ \cdot \mathbf{d}_L^*(k)\mathbf{d}_L^T(k)[\mathbf{c}_{01}^H - \mathbf{g}_2^H(k)\mathbf{D}(k-1)\mathbf{c}_1(k)]^H (35)\end{aligned}$$

where

$$\mathbf{c}_{01}^H \equiv E\{\mathbf{c}(k)\mathbf{c}^H(k-1)\} = R_c(T)\mathbf{c}_{00} \qquad (36)$$

$$\begin{aligned}\mathbf{c}_2^H(k) &\equiv E[\mathbf{c}(k+1)\mathbf{c}^H(k-1), \ldots, \mathbf{c}(k+1)\mathbf{c}^H(1\\ &= [\mathbf{c}_{00}R_c(2T), \ \ldots, \ \mathbf{c}_{00}R_c(kT)] \qquad (37\end{aligned}$$

and

$$\mathbf{g}_2^H(k) = \mathbf{c}_2^H(k)\mathbf{D}^H(k-1)\mathbf{C}_x^{-1}(k-1) \qquad (38)$$

Our goal is to seek a relationship between $\hat{\mathbf{c}}(k+1|k)$ and $\hat{\mathbf{c}}(k|k-1)$ or $\hat{\mathbf{c}}(k|k)$. Comparing Equation (26) and (38), we can see that their difference lies in $\mathbf{c}_1(k)$ and $\mathbf{c}_2(k)$. However, if the channel varies slowly with time, it can be shown that[15]

$$\mathbf{c}_2^H(k) \approx w_0\mathbf{c}_1^H(k) \qquad (39)$$

Now assume it strictly holds, the following recursive relationships can be obtained[15],

$$\begin{aligned}\hat{\mathbf{c}}(k+1|k) = w_0\hat{\mathbf{c}}(k|k) + \varepsilon_{k|k-1}^{-1}(x_k - \hat{x}_{k|k-1}) \\ \cdot (R_c(T) - w_0)\mathbf{c}_{00}\mathbf{d}_L^*(k) (40)\end{aligned}$$

and

$$\begin{aligned}\mathbf{Q}(k+1|k) = |w_0|^2\mathbf{Q}(k|k) + (1-|w_0|^2)\mathbf{c}_{00} \\ -\varepsilon_{k|k-1}|R_c(T) - w_0|^2\mathbf{c}_{00}\mathbf{d}_L^*(k)\mathbf{d}_L^T(k)\mathbf{c}_{00} \\ -2\varepsilon_{k|k-1}Re\{w_o^*(R_c(T) - w_0)\} \\ \cdot \mathbf{c}_{00}\mathbf{d}_L^*(k)\mathbf{d}_L^T(k)\mathbf{Q}(k|k-1) (41)\end{aligned}$$

where

$$\begin{aligned}\hat{\mathbf{c}}(k|k) &\equiv E\{\mathbf{c}(k)|\mathbf{x}(k), \mathbf{d}(k)\} \qquad (42)\\ \mathbf{Q}(k|k) &\equiv Cov\{\mathbf{c}(k)|\mathbf{x}(k), \mathbf{d}(k)\} \qquad (43)\end{aligned}$$

Several observations can be made about the recursive channel estimator:

1. The necessary and sufficient condition for the assumption $\mathbf{c}_2^H(k) = w_0\mathbf{c}_1^H(k)$ is

   $$R_c(jT) = w_0^j R_c(T),\ j = 2,\ 3,\ \ldots \tag{44}$$

   Any continuous-time channel whose correlation function has the following form is sufficient

   $$R_c(\tau) = w_0^{\tau/T} R_c(T) + R(\tau),\ |\tau| \geq T \tag{45}$$

   with $R(nT) = 0$.

2. If we add the condition that $R_c(T) = w_0$, then we have

   $$R_c(jT) = w_0^j,\ j = 1,\ 2,\ \ldots \tag{46}$$

   This is just the correlation function of the channel when it is a first order AR process.

3. The channel estimator is similar to the Kalman filter. Apparently, if we add one more condition that $R_c(T) = w_0$, then it is just the recursive equations of Kalman filter with the channel as the state and the received signal as observation.

4. For the isotropic mobile channel model, the difference between w_0 and $R_c(T)$ is small and both of them are very close to 1. This makes the first order AR model a reasonable approximation if the Doppler spread is small. In addition, since $w_0 \approx R_c(T) \approx 1$, it is close to a random walk model. This helps explain why the BCDFE in [6] can track the slow isotropic channel although the channel model used for channel estimator is a random-walk model.

IV. SURVIVOR-REDUCING TECHNIQUES

The optimal receiver keeps all the $|\Omega_d|^k$ sequences, i.e., all of them are survivors. This is neither practical nor necessary. Actually, the performance degradation is minimal if most of the sequences are eliminated properly, especially at high SNR. Two methods often used to reduce the number of survivors are thresholding and decision feedback [8, 6]. Decision feedback can limit the maximum number of survivors to $|\Omega_d|^K$, where K is the decision lag, while the thresholding can reduce the average number of survivors. Another method to reduce the maximum number of survivors further without changing the decision lag is Bayesian conditional feedback (BCDF), which limits the maximum number of survivors to $|\Omega_d|^J$ where $J < K$ is the Bayesian conditional decision lag. In this section, we propose a conditional decision feedback (CDF) technique, which also reduces the maximum number of survivors to $|\Omega_d|^J$, where J is the conditional decision lag. The CDF is less complicated than BCDF and also maintains the clear meaning of the conditional mean of the likelihood function as the optimal data estimate. The CDF is also compatible with decision feedback and thresholding.

Suppose we have received $\mathbf{x}(k)$ and have made conditional decision on $\mathbf{d}(k-J)$, denoted as $\bar{\mathbf{d}}(k-J)$. In the following description of the recursive CDF algorithm, we will not introduce decision feedback for simplicity. Decision feedback can be easily accommodated by replacing the $\bar{\mathbf{d}}_{k-K}()$ with final decisions $\hat{\mathbf{d}}(k-K)$. The CDF is based on the observation that the difference in decision statistics tends to be larger for the data sequences that vary in the first few symbols than for those that share the same first few. Therefore, introducing the conditional decision feedback by assuming $\forall \mathbf{d}(k+1)$

$$\begin{aligned} &f(x_{k+1}|x(k), \mathbf{d}(k+1)) = \\ &f(x_{k+1}|x(k), \mathbf{d}_{J+1}(k+1), \bar{\mathbf{d}}(k-J)) \end{aligned} \tag{47}$$

we can show that $p(\mathbf{d}_J(k+1)|\mathbf{x}(k+1))$ and $p(\mathbf{d}(k-J+1)|\mathbf{x}(k+1), \mathbf{d}_J(k+1))$ can be computed recursively from $p(\mathbf{d}_J(k)|\mathbf{x}(k))$ by the following [15]

$$\begin{aligned} &p(\mathbf{d}_J(k+1)|\mathbf{x}(k+1)) = \\ &\textstyle\sum_{d_{k-J+1}} [p(d_{k+1})p(\mathbf{d}_J(k)|\mathbf{x}(k)) \\ &\cdot f(x_{k+1}|\mathbf{x}(k), \mathbf{d}_{J+1}(k+1), \bar{\mathbf{d}}(k-J))] \\ &\cdot [p(\mathbf{x}(k+1))/p(\mathbf{x}(k))]^{-1} \end{aligned} \tag{48}$$

with

$$\begin{aligned} &p(\mathbf{x}(k+1))/p(\mathbf{x}(k)) = \\ &\textstyle\sum_{\mathbf{d}_{J+1}(k+1)} p(d_{k+1})p(\mathbf{d}_J(k)|\mathbf{x}(k)) \\ &\cdot f(x_{k+1}|\mathbf{x}(k), \mathbf{d}_{J+1}(k+1), \bar{\mathbf{d}}(k-J)) \end{aligned} \tag{49}$$

and

$$\begin{aligned} & p(\mathbf{d}(k-J+1)|\mathbf{x}(k+1),\mathbf{d}_J(k+1)) = \\ & p(\mathbf{d}(k-J)|\mathbf{x}(k),\mathbf{d}_J(k))p(d_{k+1}) \\ & \cdot f(x_{k+1}|x(k),\mathbf{d}_{J+1}(k+1),\bar{\mathbf{d}}(k-J)) \\ & \cdot [p(\mathbf{d}_J(k+1)|\mathbf{x}(k+1))\tfrac{p(\mathbf{x}(k+1))}{p(\mathbf{x}(k))}]^{-1} \end{aligned} \tag{50}$$

with

$$\begin{aligned} & p(\mathbf{d}_J(k+1)|\mathbf{x}(k+1))\tfrac{p(\mathbf{x}(k+1))}{p(\mathbf{x}(k))} = \\ & \textstyle\sum_{d_{k-J+1}} p(d_{k+1})p(\mathbf{d}_J(k)|\mathbf{x}(k)) \\ & \cdot f(x_{k+1}|\mathbf{x}(k),\mathbf{d}_{J+1}(k+1),\bar{\mathbf{d}}(k-J)) \end{aligned} \tag{51}$$

Obviously, the number of survivors is at most $|\Omega_d|^J$, since there are only $|\Omega_d|^J$ possible $p(\mathbf{d}_J(k)|\mathbf{x}(k))$.

The conditional decision on $\mathbf{d}(k-J+1)$ is given by

$$\begin{aligned} & \bar{\mathbf{d}}(k-J+1) = \arg\max\nolimits_{\mathbf{d}(k-J+1)} \\ & p(\mathbf{d}(k-J+1)|\mathbf{x}(k+1),\mathbf{d}_J(k+1)) \end{aligned} \tag{52}$$

and the final decision on d_{k-K+1}

$$\begin{aligned} & \hat{d}_{k-K+1} = \arg\max\nolimits_{d_{k-K+1}} \textstyle\sum_{\mathbf{d}_J(k+1)} \\ & p(\mathbf{d}(k-J+1)|\mathbf{x}(k+1),\mathbf{d}_J(k+1)) \\ & \cdot p(\mathbf{d}_J(k+1)|\mathbf{x}(k+1)) \end{aligned} \tag{53}$$

Note that the conditional decision feedback introduced here is only on the distribution function, weaker than the Bayesian conditional decision feedback equalizer (BCDFE). BCDFE assumes that $\mathbf{d}(k-J)$ is a function of $\mathbf{d}_J(k)$ and $\mathbf{x}(k)$. This results in more involved computation of $\hat{x}_{k|-1}$ and $\varepsilon_{k|k-1}$, which also lose the clear meaning that $\hat{x}_{k|-1}$ is the LPF output for a given data sequence. Just like decision feedback, conditional decision feedback lags limit the maximum number of survivors. Thus it determines the memory size.

V. SIMULATION RESULTS AND CONCLUSIONS

In this section we present the results of our simulation study of the algorithms proposed in this paper. We denote the algorithm with the LPF length N and decision feedback as the MAP(N) algorithm. We used an two-ray and symbol-spaced fading channel model with equal gain for each path and Jakes' model [13] to generate the isotropic fading process. We also assumed known timing and noise variance for the receiver. When PSAM is used, the pilot symbol spacing is denoted PSS.

The parameters affecting the performance of the MAP(N) algorithm include the LPF length N, the decision lag K, and the threshold value (TP). The estimated bit error rate (BER) for BPSK modulation with different values of N are shown in Fig. 1 with $TP = 10^{-37}$, $K = 6$ and $PSS = 5$ for both the MAP(N) algorithm and the ideal coherent BPSK (i.e., exactly known channel). The performance is a monotonically increasing function of N but for large enough N the additional gain for each increase is relatively minor. The performance of the algorithm is also maintained over a wide range of PSS as shown in Fig. 2. Thresholding decision statistics can significantly reduce the complexity. The effectiveness of thresholding is seen in Figure 3 and Figure 4, where the BERs and the average number of survivors are plotted respectively using different TPs with $K = 6$,$N = 16$ and $PSS = 5$. These figures show that thresholds can be chosen which maintain near ideal performance while greatly reducing the complexity.

The effectiveness of conditional decision feedback (CDF) can be seen in Figure 6, where the performances with or without CDF are compared with the same maximum number of survivors. With conditional decision lag $J = 3$, the performance is close to that without CDF while the number of survivors is by 7/8.

The advantage of MAP(N) with CDF in performance over BCDFE and feedforward method is clearly shown in Figure 7. For the feedforward, we insert a training sequence of length 5 every 15 informational symbols, so that we can use the self-orthogonal sequence [2, 14] and sampling rate is higher than $2f_D$. Note that the feedforward has used more training symbols than MAP(N). Actually, MAP(N) does not need any pilot symbols, as shown in Figure 8, where the performances of MAP(N) with $PSS = 5$ or without pilot symbols are compared.

In summary, the performance of these suggested algorithms is close to that of the ideally coherent but unachievable receiver and outperforms other previously proposed algorithms. In addition, they do not require training sequences and are robust against mismatch in Doppler spread. However, further study is needed to reduce the complexity of these algorithms.

REFERENCES

[1] J. K. Cavers,"An analysis of pilot symbol assisted modulation for Rayleigh fading channels," *IEEE Trans. Vehicular Technol.*, vol. 40, pp.686-693, Nov. 1991.

[2] S. A. Fechtel and H. Meyr,"Optimal parametric feedforward estimation of frequency-selective fading radio channels," *IEEE Trans. Commun.*, vol. 42, pp. 1639-1650, Feb./March/April 1994.

[3] R. Raheli, A. Polydoros and C. Tzou,"Per-survivor processing," *Digital Signal Processing-A Review Journal*, Academic Press, vol.3, pp.175-187, July 1993.

[4] J. Lin, F. Ling and J. G. Proakis,"Joint data and channel estimation for TDMA mobile channels," *Intern. J. Wireless Inform. Net.*, vol.1, pp.229-238, No.4, 1994.

[5] R. A. Iltis,"A Bayesian maximum-likelihood sequence estimation algorithm for *a priori* unknown channels and symbol timing," *IEEE Trans. J. Selec. Areas Commun.*, vol. 10, pp.579-589, April 1992.

[6] G. Lee, M. P. Fitz and S. B. Gelfand,"Bayesian techniques for equalization of rapidly fading frequency selective Channels," *Intern. J. Wireless Inform. Net.*, vol. 2, pp.41-54, Jan. 1995.

[7] J. H. Lodge and M. L. Moher, "Maximum likelihood sequence estimation of CPM signals transmitted over Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 38, pp. 787-794, June 1990.

[8] J. P. Seymour and M. P. Fitz, "Near-optimal symbol-by-symbol detection schemes for flat Rayleigh fading," *IEEE Trans. Commun.*, vol. 43, pp. 1525-1534, Feb./March/April 1994.

[9] N. Seshadri and J. H. Winters, "Two signaling schemes for improving the error performance of frequency division duplex(FDD) transmission systems using transmitter antenna diversity," *Intern. J. Wireless Inform. Net.*, vol. 1, pp. 49-59, No. 1, 1994.

[10] A. Wittneben, "A new bandwidth efficient transmit antenna modulation diversity scheme for linear digital modulation," *IEEE ICC'93*, Geneva, Switz., May 1993, pp.1630-1634.

[11] J. Cavers, "On the validity of the slow and moderate fading models for matched filter detection of Rayleigh fading signals," *Canadian Journal of Electrical and Computer Engineering*, vol. 17, pp. 183-189, Oct. 1992.

[12] J. G. Proakis, *Digital Communications'*, McGraw-Hill, New York, 1989.

[13] W. C. Jakes, *Microwave Mobile Communications*, Wiley, New York, 1974.

[14] A.P.Clark, Z.C.Zhu and J.K.Joshi,"Fast start-up channel estimation," em IEE Proc. vol.131, pp.375-382, July 1984.

[15] Y. Zhang,"Optimal and near-optimal joint channel and data estimation for Rayleigh fading channels," Preliminary proposal for Ph.D, Purdue University, 1995.

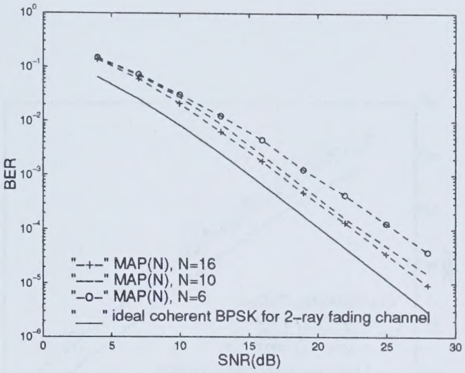


Figure 1: The performance of MAP(N) algorithm with different LPF lengths for the two-ray fading channel with $f_D = 0.02, PSS = 5, K = 6$

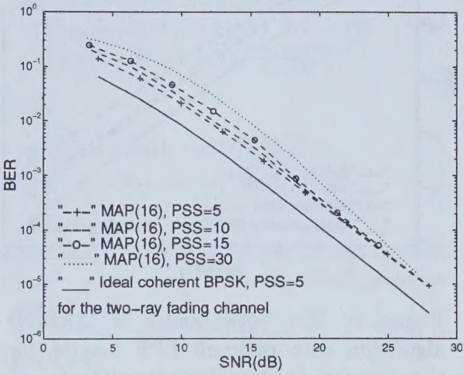


Figure 2: The performance of MAP(N) algorithm with different PSS for the two-ray fading channel with $f_D = 0.02, N = 16, K = 6$

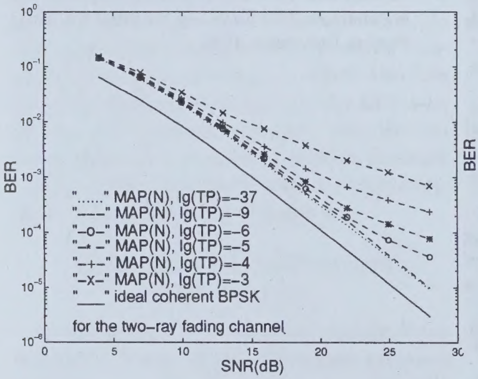


Figure 3: The performance of MAP(N) algorithm with different TPs for the two-ray fading channel with $f_D = 0.02, N = 16, K = 6, PSS = 5$

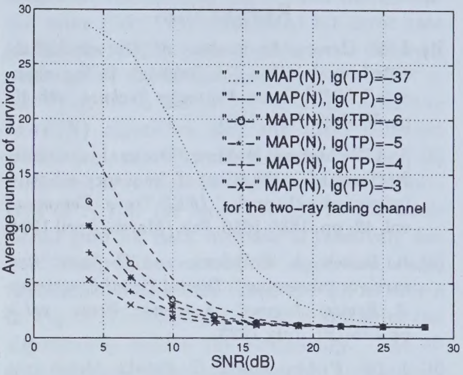


Figure 4: The average number of survivors of MAP(N) algorithm for two-ray fading channel with $f_D = 0.02, N = 16, K = 6, PSS = 5$

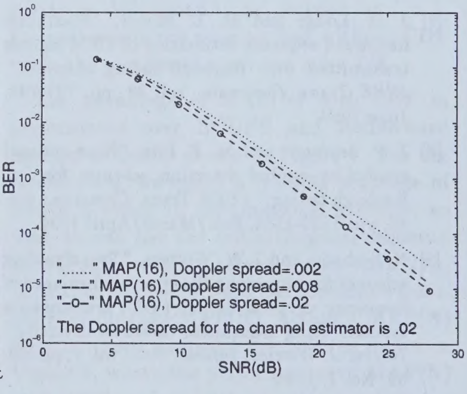


Figure 5: The BER of the MAP(N) algorithm with its Doppler spread greater than that of the actual channel under the two-ray fading channel with $N = 16$, $K = 6$, $PSS = 5$, $\lg(TP) = -37$

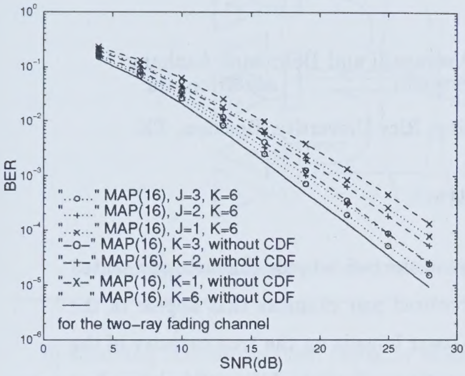


Figure 6: Comparison of MAP(N) algorithms with or without conditional decision feedback for BPSK under the two-ray fading channel with $f_D = 0.02$, $N = 16$, $PSS = 5, \lg(TP) = -37$

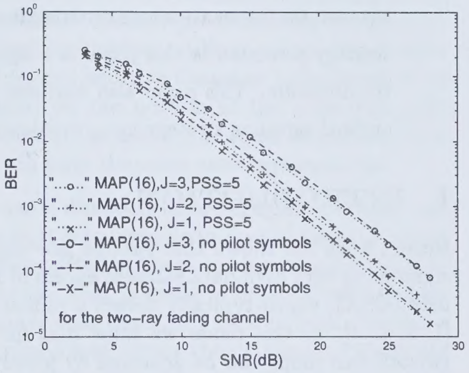


Figure 8: Comparison of MAP(N) algorithm with and without pilot symbols for the two-ray fading channel with conditional decision lag J.

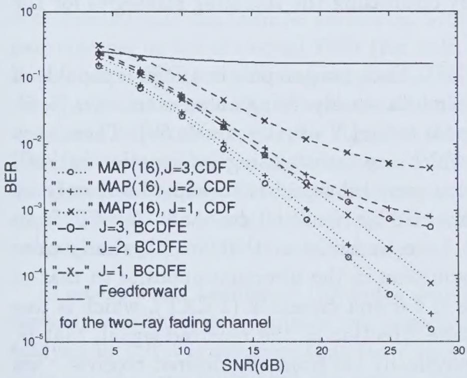


Figure 7: Comparison of the MAP(16) with BCDFE and feedforward for BPSK under the two-ray fading channel with $f_D = 0.02$, $K = 6$, $PSS = 5$, $\lg(TP) = -37$

Joint Signaling Strategies for Maximizing the Capacity of Twisted Pair Loops

Andrew Sendonaris, Venugopal Veeravalli and Behnaam Aazhang
sendos@rice.edu venu@rice.edu aaz@rice.edu

Department of Electrical Engineering, Rice University, Houston, TX.

Abstract

A method for approaching the capacity of the twisted pair channel is presented and analyzed. It is then applied to two twisted pair channels that appear in the literature. Although all results represent lower bounds on the true capacity of the channel, the information rate resulting from the application of the optimal signaling strategy presented in this paper is a significant improvement over what exists in the literature. This paper also discusses a straightforward implementation of the optimal signaling strategy using multicarrier modulation.

I INTRODUCTION

Recent work has shown that *twisted pair* digital subscriber loops (DSL's) are capable of supporting very high data rates–in excess of 1.6 Mb/s within the standard *carrier serving area* (CSA), which typically covers a radius of about 12,000 ft around the central office [1, 2, 3, 4]. In this paper we show that further significant increases in the capacity of twisted pair loops can be achieved by jointly optimizing the signaling strategies for the two directions of transmission on the loops.

The following is a description of a typical DSL. Each twisted pair in a DSL is capable of supporting transmissions in both directions simultaneously, using an echo canceler [5, 6]. Suppose there are N twisted pairs in one binder cable (N usually equals 50). Then, each twisted pair has $N-1$ neighboring pairs which are transmitting information in both directions. At each end of a twisted pair, the received signal is corrupted not only by Additive White Gaussian Noise (AWGN), but also by Near-end crosstalk (NEXT) from the $N-1$ neighboring twisted pairs, which have transmitters that are physically close to the desired receiver, and are sending information in the direction opposite to that of the received signal. Another source of noise is Far-end crosstalk (FEXT), which is due to the transmission of information in the same direction as the received signal, that is, information sent by transmitters that are physically far from the desired receiver. See Figure 1.

In ADSL (Asymmetrical Digital Subscriber Line) systems, all twisted pairs transmit signals in the same direction, thus eliminating NEXT [4, 7]. Therefore, when studying ADSL, FEXT is the only source of crosstalk and thus needs to be considered. However, the effects of FEXT are negligible whenever NEXT is present in the channel [4, 3], as is the case with all systems that support full-duplex transmission, such as HDSL (High bit rate Digital Subscriber Lines). In this paper we will focus on the latter full-duplex DSL's.

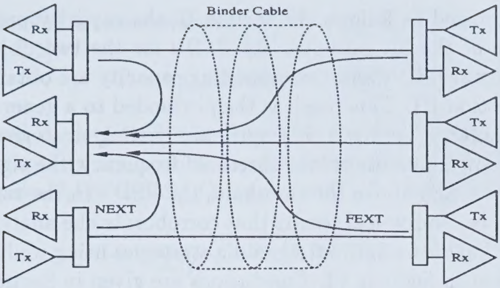


Figure 1: Schematic Representation of NEXT and FEXT in a DSL

In studying the information capacity of DSL's, all the twisted pairs in a binder group can be considered jointly as a multiuser interference channel [8]. Obtaining the capacity region[1] of such a multiuser interference channel is, however, still an open problem even for the simplest case of two users, discrete time and additive Gaussian background noise [8]. To make the problem tractable, we focus on the portion of the capacity region corresponding to equal information rates for each of the users. The equal rate assumption is valid in the HDSL context since all users will have the same rate requirements.

Since signals that are transmitted in the same direction have negligible crosstalk between them, in the maximization of the bit rate, it is reasonable to assume that the power spectral densities (PSD's) of all the signals transmitted in the same direction are equal to the same optimal PSD. An additional assumption that is made in much of the previous work on capacity calculations for twisted pairs is that the PSD's used in both directions of transmission are identical [9, 3]. The optimization to obtain the maximum bit rate for each twisted pair can then be performed over a single PSD. The capacity of each twisted pair channel under the equal PSD (for both directions of transmission) constraint, and the corresponding maximizing signal PSD have been derived [9, 3]. However, as noted in these papers, the capacity of each twisted pair under the equal PSD constraint converges to a finite constant value as the signal to (background) noise ratio (SNR) goes to infinity. This fact is an indication that the equal PSD constraint is too restrictive. If, for example, the signals in the two directions of transmissions are chosen to have PSD's that occupied disjoint frequency bands in the available bandwidth, the capacity of each twisted pair is unbounded as the SNR goes to infinity.

In recent work, Lechleider [10] notes that the equal PSD constraint leads to decreased capacity and further calculates the capacity without this constraint. However, in the analysis in [10] it is assumed that the PSD's have the same support–an assumption that rules out the possibility that the signals have disjoint spectra. The results given in [10] show only a slight improvement in capacity over equal PSD case.

In this paper, we show that significant gains in capacity can be obtained by removing the equal PSD constraint. The capacity obtained with the optimal signaling strategy can be far larger than the equal PSD capacity; furthermore, it goes to infinity as the SNR approaches infinity.

---

[1] The capacity region of a multiuser channel is the set of all possible combinations of achievable user rates.

This paper is organized as follows. In Section II, the capacity problem is formulated as the maximization of the bit rate over the PSD's for the two directions of transmission. The optimal signal PSD's and corresponding capacity are obtained for a simplified channel model in Section III. This result is then extended to a general channel model in Section IV. In particular, it is shown that optimal signaling strategies have the following simple structure: Below a predetermined threshold frequency, the signal PSD's are identical in both directions; and above the threshold, the PSD's in the two directions occupy disjoint frequency slots. Numerical results that corroborate the analysis are given in Section V. Implementation of the optimal signaling strategies using multicarrier modulation techniques is discussed in Section VI. Conclusions are given in Section VII.

II PROBLEM FORMULATION

As mentioned in Section I, we assume that each direction of transmission will have its own PSD, say $S_1(f)$ and $S_2(f)$. Let direction 1 refer to the direction of transmission using $S_1(f)$ as its PSD, and let direction 2 refer to the other direction.

To obtain an analytical expression for the capacity of the twisted-pair channel, some assumptions are made. The channel is assumed to have transfer function $H_c(f)$, and two sources of noise: AWGN (Additive White Gaussian Noise) with one-sided spectral density N_o, and NEXT (Near-end crosstalk) with crosstalk transfer function $H_x(f)$. The crosstalk noise is assumed to be Gaussian. To find capacity, crosstalk can be modeled as Gaussian noise without substantial error [11]. Under such assumptions, the capacity for direction 1 is given by [9, 3, 10]

$$C_1 = \sup_{S_1(f), S_2(f)} \int_0^\infty \log_2 \left[1 + \frac{|H_c(f)|^2 S_1(f)}{N_o + |H_x(f)|^2 S_2(f)} \right] df \;\; (bits/sec) \tag{1}$$

where the supremum is taken over all $S_1(f)$ and $S_2(f)$ such that

$$S_i(f) \geq 0 \;\; \forall f \text{ , and } \int_0^\infty S_i(f) df \leq P_{\max} \;\; \text{for } i = 1, 2$$

and where $S_i(f)$ is the PSD of direction i, and the latter inequality corresponds to an average power constraint.

The capacity for direction 2 is given by the analogous expression. Of course, when designing the transmission scheme for both directions, it becomes necessary to consider them both simultaneously. As can be seen from (1), when only direction 1 is considered, C_1 is achieved when $S_2(f) = 0 \;\; \forall f$, which implies that $C_2 = 0$. The same problem arises when only direction 2 is considered. Therefore, a joint optimization needs to be performed. There are several ways in which this can be done.

Let

$$C^*(S_1(f), S_2(f)) = \int_0^\infty \log_2 \left[1 + \frac{|H_c(f)|^2 S_1(f)}{N_o + |H_x(f)|^2 S_2(f)} \right] df$$

Then

$$C_1 = C^*(S_1^*(f), S_2^*(f)) \text{ and } C_2 = C^*(S_2^*(f), S_1^*(f))$$

where $S_1^*(f)$ and $S_2^*(f)$ are chosen to maximize C_1 subject to the constraint that $C_1 = C_2$, or to maximize $[C_1 + C_2]$, or maximize $\min[C_1, C_2]$.

All of the above approaches are analytically intractable. An approach to solving for C_1 and C_2 that appears in much of the literature, is to assume that the PSD's are the same in both directions of transmission, i.e. $S_1(f) = S_2(f)$. Under this assumption, the problem reduces to [9, 3]

$$C_1 = \max_{S_1(f)} C^*(S_1(f), S_1(f)) \tag{2}$$

However, as discussed in Section I, the equal PSD assumption is too restrictive, and significant gains in capacity are possible by removing this assumption. The main idea of this paper is to make $S_1(f)$ and $S_2(f)$ "symmetric", in some sense, thus reducing the problem to

$$C_1 = \max_{S_1(f)} C^*(S_1(f), S_1^{\text{sym}}(f)) \tag{3}$$

where $S_1^{\text{sym}}(f)$ is symmetric to $S_1(f)$. Despite the fact that it is difficult to rigorously define the concept of symmetry, generally speaking it implies that $S_1^{\text{sym}}(f)$ is small where $S_1(f)$ is large, and vice versa, so that $C_1 = C^*(S_1(f), S_1^{\text{sym}}(f)) = C^*(S_1^{\text{sym}}(f), S_1(f)) = C_2$. This concept will become clearer in the following section.

The second idea is to parameterize $S_1(f)$ so that the maximization is taken over the parameters that now describe $S_1(f)$

III SIMPLIFIED CHANNEL

Consider a channel with

$$|H_c(f)|^2 = \begin{cases} H & \text{if } |f| \leq \text{W} \\ 0 & \text{otherwise} \end{cases}, \quad |H_x(f)|^2 = \begin{cases} X & \text{if } |f| \leq \text{W} \\ 0 & \text{otherwise} \end{cases}$$

Consider the class of $S_1(f)$ and $S_2(f)$ defined as follows:

$$S_1(f) = \begin{cases} \alpha \frac{2P_{\max}}{W} & \text{if } |f| \leq \frac{W}{2} \\ (1-\alpha)\frac{2P_{\max}}{W} & \text{if } \frac{W}{2} < |f| \leq W \\ 0 & \text{otherwise} \end{cases}, \quad S_2(f) = \begin{cases} (1-\alpha)\frac{2P_{\max}}{W} & \text{if } |f| \leq \frac{W}{2} \\ \alpha \frac{2P_{\max}}{W} & \text{if } \frac{W}{2} < |f| \leq W \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $0.5 \leq \alpha \leq 1$. In this configuration, $\alpha = 0.5$ corresponds to using equal PSD's for the two directions, and $\alpha = 1$ corresponds to using Frequency Division Signaling (FDS). By construction of $S_1(f)$ and $S_2(f)$, $C_1 = C^*(S_1(f), S_2(f)) = C^*(S_2(f), S_1(f)) = C_2$. Therefore, $S_1(f)$ and $S_2(f)$, as defined above, are symmetric in the sense described in Section II. The resulting capacity for direction 1 becomes

$$C_1 = \max_{0.5 \leq \alpha \leq 1} \frac{W}{2\ln(2)} \left\{ \ln\left[1 + \frac{\alpha S H}{1 + (1-\alpha) S X}\right] + \ln\left[1 + \frac{(1-\alpha) S H}{1 + \alpha S X}\right] \right\}$$

where $S = \frac{2P_{\max}}{N_o W}$. Due to the symmetry of $S_1(f)$ and $S_2(f)$, C_2 will be equal to C_1 for any value of α. Therefore, it is only necessary to maximize C_1. Taking the derivative with respect to α,

$$\frac{\partial C^*}{\partial \alpha} = (2\alpha - 1)(2X - H + SX^2)Q$$

where $Q = \frac{W}{2\ln(2)} \frac{SH(1+SX)}{(1+(1-\alpha)SX+\alpha SH)(1+(1-\alpha)SX)(1+\alpha SX+(1-\alpha)SH)(1+\alpha SX)} > 0 \quad \forall \alpha \in [0.5, 1]$. Therefore, $\alpha = \frac{1}{2}$ is the only stationary point. If it is a maximum, it is optimum to use equal PSD's for the two directions of transmission. If it is a minimum, then $\alpha = 1$ achieves the maximum, which means that it is optimum to use Frequency Division Signaling for the two directions.

For all $\alpha > 0.5$, the derivative of C^* with respect to α will be negative if and only if $2X - H + SX^2 < 0$. This implies that $\alpha = \frac{1}{2}$ is a maximum if and only if $S < \frac{H-2X}{X^2}$. Similarly, $\alpha = \frac{1}{2}$ is a minimum if and only if $S > \frac{H-2X}{X^2}$. Thus, since $S = \frac{2P_{\max}}{N_oW}$, it follows that

$$\frac{2P_{\max}}{N_oW} \underset{\text{fds}}{\overset{\text{eqpsd}}{\lessgtr}} \frac{H-2X}{X^2} \tag{5}$$

where *eqpsd* denotes using equal PSD's for the two directions of transmission and *fds* denotes using Frequency Division Signaling. Although the result in (5) was obtained under the assumption that $S_1(f)$ and $S_2(f)$ have the form described in (4), there are indications that achieving the global maximum always involves using the same threshold test as in (5).

IV GENERALIZED CHANNEL

Consider a channel that is bandlimited to $|f| \leq W$, however no longer restricted to being constant over this frequency range. The frequency range $0 \leq f \leq W$ is divided into M equal-width bins (M parallel independent subchannels). Within each bin, both $H_c(f)$ and $H_x(f)$ will be approximately constant, given that M is sufficiently large. Also, any capacity-maximizing scheme will have to decide how much power to allocate for each bin. Therefore, using the results from the previous section, it follows that in the i^{th} bin

$$\frac{2P_i}{N_oD} \underset{\text{fds}}{\overset{\text{eqpsd}}{\lessgtr}} \frac{H_i-2X_i}{X_i^2} \tag{6}$$

where $D = \frac{W}{M}$, $H_i = H_c(f_i)$, $X_i = H_x(f_i)$, $f_i = \frac{D}{2} + (i-1)D$. ($f_i$ is the center frequency of the i^{th} bin)

Assume $H_c(f)$ is a decreasing function of f and $H_x(f)$ is an increasing function of f, as is the case for twisted pair channels. This implies that $\frac{H_c(f)-2H_x(f)}{H_x^2(f)}$ is a decreasing function of f. Therefore, $\frac{H_i-2X_i}{X_i^2}$ is a decreasing function of i, which implies that for some i^*, $\frac{H_i-2X_i}{X_i^2} < 0 \ \forall i \geq i^*$. Since $P_i \geq 0 \ \forall i$, it follows that $\frac{2P_i}{N_oD} > \frac{H_i-2X_i}{X_i^2} \ \forall i \geq i^*$, which in turn implies that Frequency Division Signaling is always optimum for all bins after (and including) the i^{*th} bin, according to (6).

Therefore, the capacity-maximizing scheme must use Frequency Division Signaling $\forall i \geq i^*$. Assume that as i decreases from i^* to 1, the strategy changes from Frequency Division to equal PSD's. That is, $\exists M_o$, in the range $1 \leq M_o < i^*$, such that equal PSD's are used in the M_o^{th} bin and Frequency Division is used in all bins to the right of the M_o^{th} bin. Theorem 1 shows that the capacity-maximizing scheme will use equal PSD's for all bins from 1 to M_o, and Frequency Division Signaling for bins $M_o + 1$ to M.

Lemma 1 *Let*

$$|H_c(f)|^2 = \begin{cases} H_1 & \text{if } 0 \leq |f| \leq \frac{W}{2} \\ H_2 & \text{if } \frac{W}{2} < |f| \leq W \\ 0 & \text{otherwise} \end{cases}, \quad |H_x(f)|^2 = \begin{cases} X_1 & \text{if } 0 \leq |f| \leq \frac{W}{2} \\ X_2 & \text{if } \frac{W}{2} < |f| \leq W \\ 0 & \text{otherwise} \end{cases}$$

Assume that there is a power constraint on the transmitted signal. Also assume that $H_i - 2X_i > 0$ for $i = 1, 2$. Then, if $H_1 > H_2$ and $X_1 < X_2$, it is always better to use equal PSD's in bin 1 and Frequency Division Signaling in bin 2, than to use FDS in bin 1 and equal PSD's in bin 2.

The proof of Lemma 1 is lengthy and involved and does not permit inclusion in this paper. However, the proof can be found in a Technical Report at Rice University [12].

Theorem 1 *Let $H_c(f)$ be a decreasing, and $H_x(f)$ an increasing function of f. Consider the class of all M-bin transmission schemes, that is, schemes that consist of dividing the given frequency range into M bins and within each bin deciding to perform equal PSD's or Frequency Division Signaling. The capacity-maximizing scheme will use equal PSD's for bins 1 to M_o, and Frequency Division for bins $M_o + 1$ to M, for some $0 \leq M_o \leq M$*

Proof: First, it is convenient to make a few notational definitions. Let e_i indicate that equal PSD's will be used in bin i, and f_i indicate that Frequency Division Signaling will be used in bin i. As was argued above, the capacity-maximizing scheme will have the following form: $\ldots, e_{M_o}, f_{M_o+1}, f_{M_o+2}, \ldots, f_M$, for some $0 \leq M_o \leq M$. The transmission strategies for bins $M_o - 1$ to 1 are not yet known. Assume that as i decreases from M_o to 1, the strategy changes from equal PSD's to Frequency Division Signaling. That is, assume that for some $i_x, 1 \leq i_x < M_o$, the following is true about the capacity-maximizing scheme

$$\ldots, f_{i_x}, e_{i_x+1}, e_{i_x+2}, \ldots, e_{M_o-1}, e_{M_o}, f_{M_o+1}, f_{M_o+2}, \ldots, f_M \tag{7}$$

Then, capacity can always be increased by switching the transmission strategy in bins i_x and M_o, while keeping the total power in both bins the same. That is, the following scheme: $\ldots, e_{i_x}, e_{i_x+1}, e_{i_x+2}, \ldots, e_{M_o-1}, f_{M_o}, f_{M_o+1}, f_{M_o+2}, \ldots, f_M$ has a higher capacity due to Lemma 1. However, this leads to a contradiction because the scheme described in (7) was assumed to maximize capacity. Thus, there is no $i_x, 1 \leq i_x < M_o$ such that the transmission strategy changes from equal PSD's to Frequency Division Signaling, which implies that the capacity-maximizing scheme will have the following form: $e_1, e_2, \ldots, e_{M_o-2}, e_{M_o-1}, e_{M_o}, f_{M_o+1}, f_{M_o+2}, \ldots, f_M$. Q.E.D.

This result is intuitively reasonable: Let both directions of transmission utilize the "good" parts of the channel (where crosstalk is low), and in the parts where the crosstalk becomes a problem use frequency division to eliminate it.

Therefore,

$$C(P_{\max}) = \max_{\substack{M_o, P_1, P_2 \\ s.t.\ P_1 + P_2 = P_{\max}}} C_{\text{eqpsd}}(M_o, P_1) + C_{\text{fds}}(M_o, P_2) \tag{8}$$

where $C_{\text{eqpsd}}(Mo, P_1)$ is the capacity of the channel when equal PSD's are used for both directions of transmission, and when the channel is bandlimited to M_oD (Hz) and the power constraint is P_1. Also, $C_{\text{fds}}(M_o, P_2)$ is the capacity of the channel when Frequency Division Signaling is used for both directions of transmission, and when the channel is bandlimited from M_oD to W (Hz) and the power constraint is P_2. In this configuration, $C_{\text{fds}}(M_o, P_2)$ corresponds to the capacity of a channel with only AWGN and with a transfer function that is an altered version of $H_c(f)$, formed by taking away the parts of the spectrum that have been assigned to the other direction of transmission.

Both $C_{\text{eqpsd}}(Mo, P_1)$ and $C_{\text{fds}}(M_o, P_2)$ are well-studied and well-described in the literature [9, 3]. However, they have no closed-form expression. Therefore, the maximization cannot be performed analytically, only numerically. Note that $C(P_{\max})$ is only a lower bound on the true capacity of the channel since we are constraining the form of $S_1(f)$ and $S_2(f)$.

V NUMERICAL RESULTS AND DISCUSSION

The capacity $C(P_{\max})$ was numerically calculated for two twisted-pair channels; one studied in [9] and one studied in [3].

1. The model of the twisted-pair channel used in [9] was

$$|H_c(f)|^2 = \exp\left(-\alpha\sqrt{f}\right) \ , \quad |H_x(f)|^2 = \beta f^{\frac{3}{2}}$$

where $\alpha = k\frac{l}{l_o}$, l =length of channel in ft, l_o =reference length= 18,000 ft, k =constant of the physical channel= 1.158, and $\beta = 10^{-9}$. In the above expressions, f refers to frequency in KHz. The results obtained (for $l = 600$) are shown in Figure 2.

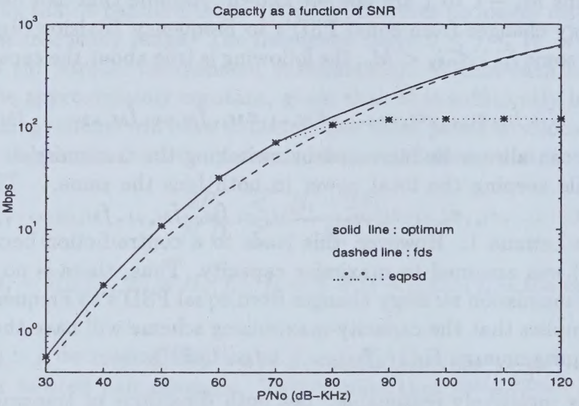


Figure 2: Capacity calculations for twisted-pair model from [9]: solid line corresponds to using the optimum transmission scheme (as described in Section IV), star-dot line corresponds to using equal PSD's for the two directions of transmission (as assumed in [9]), and dashed line corresponds to using Frequency Division Signaling in all bins

---

2. For the model of the twisted-pair channel used in [3],

$$|H_c(f)|^2 = \theta \exp\left(-\alpha\sqrt{f}\right) \ , \quad |H_x(f)|^2 = K f^{\frac{3}{2}} \tag{9}$$

where α is defined as in the previous model, θ represents the attenuation at $f = 0$, and $K = 10^{-13}$. In the above expressions, f refers to frequency in Hz.

It is important to note that a closed form expression for $|H_c(f)|^2$ was not given in [3]. However, the parameters α and θ, in (9), were selected so that the resulting $|H_c(f)|^2$ matched the plot given in [3] as closely as possible. The fact that the model of $|H_c(f)|^2$ used in this example is not *exactly* equal to the one used in [3] is not important, since it is the approach to achieving capacity that is being tested, and this can accomplished using any model for $|H_c(f)|^2$. The results obtained are shown in Figure 3

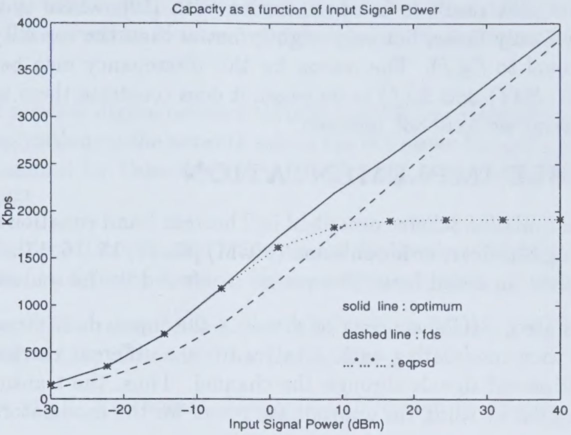


Figure 3: Capacity calculations for twisted-pair model from [3]: solid line corresponds to using the optimum transmission scheme (as described in Section IV), star-dot line corresponds to using equal PSD's for the two directions of transmission (as assumed in [3]), and dashed line corresponds to using Frequency Division Signaling in all bins

What is plotted in both Figures 2 and 3 is the capacity under three different transmission schemes. One uses equal PSD's for all bins (same as in [9]), one uses Frequency Division in all bins, and the other is the optimal, that is, it uses equal PSD's for the first M_o bins (for some $0 \leq M_o \leq M$) and Frequency Division for the remaining bins. These correspond to the plots of $C_{\text{eqpsd}}(M, P_{\max})$, $C_{\text{fds}}(0, P_{\max})$ and $C(P_{\max})$, respectively, versus SNR.

There are a few observations that can be made on both figures. First, $C(P_{\max})$ is larger than the other two for all values of SNR, as should be expected. More importantly, $C(P_{\max})$ *always increases* with SNR, unlike $C_{\text{eqpsd}}(M, P_{\max})$ which has a maximum value no matter how large the SNR becomes, as was also observed by [9] and [3].

Also, for low values of SNR $C(P_{\max})$ approaches $C_{\text{eqpsd}}(M, P_{\max})$, and for high values of SNR $C(P_{\max})$ approaches $C_{\text{fds}}(0, P_{\max})$. In fact, for low SNR the optimum M_o in equation (8) approaches M, and for high SNR it approaches 0. That is, for low SNR it is optimum to use equal PSD's in all bins, and for high SNR it is optimum to use Frequency Division Signaling in all bins. This result should be expected because at low SNR, crosstalk is not an important source of interference, implying that both directions of transmission can use all the frequency spectrum. On the other hand, at very high SNR, crosstalk becomes a problem at all frequencies, implying that it should be eliminated using Frequency Division.

It is important to discuss the results of [10], because it is the only work that questions the assumption that $S_1(f)$ has to be equal to $S_2(f)$ when deriving capacity of a twisted-pair channel. In [10], it is assumed that $N_o = 0$, that is, the only source of noise is NEXT. From (6) it follows that if the frequency range $0 \leq f \leq W$ is divided into M bins, then Frequency Division Signaling *has* to be used in all bins. Thus, each direction of transmission faces a channel with no noise. The resulting capacity is infinite.

It appears that this result is in contradiction with [10], where it was shown that the capacity is not only finite, but only slightly better than the capacity when $S_1(f)$ is assumed to be equal to $S_2(f)$. The reason for this discrepancy may be that while [10] does not constrain $S_1(f)$ and $S_2(f)$ to be equal, it does constrain them to have the same support, a constraint we have not imposed.

VI POSSIBLE IMPLEMENTATION

The optimum transmission scheme described in Theorem 1 and equation (8) can be easily implemented using Multicarrier Modulation (MCM) [13, 14, 15, 16, 17]. The modulation will not be explained in detail here. The reader is referred to the sources cited above.

Very briefly stated, MCM consists of dividing the input data stream into parallel substreams and then modulating each substream on a different carrier frequency and sending the multiplexed signals through the channel. Thus, the transmitter can shape the PSD of the signal by using the appropriate power for the modulators at each carrier frequency.

Therefore, it is straightforward, using Multicarrier Modulation, to implement the proposed optimal signaling scheme. The first M_o carriers should be assigned to both directions of transmission, while the remaining carriers should be assigned alternatingly to each direction. Also, the appropriate power (and bit rate) should be used for each carrier so that the signal PSD is as desired. It is important to note that the transmitted signal needs to be as Gaussian as possible, so that (1) is valid.

VII CONCLUSIONS

We established that allowing different PSD's for the two directions of transmission over twisted pair channels and performing a joint optimization over these PSD's can result in significant increases in the information capacity. In particular, we showed that the capacity is unbounded as the SNR goes to infinity. The optimal signaling strategies have a simple structure, that lends itself to an implementation using multicarrier modulation.

An interesting topic for further research is the analysis of the actual bit rate achievable using a multicarrier QAM signaling scheme that uses the optimal PSD's derived in this paper. It is to be noted that, in deriving capacity, this paper assumed a constraint on the average power. Another avenue for further research is the capacity analysis under peak power constraints, akin to the analysis in [18], however without the equal PSD constraint. Also, since the analysis in this paper does not apply directly to ADSL, another possibility for future work would be the application of the approach used in this paper to ADSL systems.

References

[1] J. W. Lechleider, "High bit rate digital subscriber lines: A review of HDSL progress," *IEEE Journal on Selected Areas in Comm.*, vol. 9, Aug. 1991.

[2] J.-J. Werner, "The HDSL environment," *IEEE Journal on Selected Areas in Communications*, vol. 9, Aug. 1991.

[3] J. T. Aslanis and J. M. Cioffi, "Achievable information rates on digital subscriber loops: Limiting information rates with crosstalk noise," *IEEE Trans. on Commun.*, vol. 40, Feb. 1992.

[4] D. L. Waring, "The asymmetrical digital subscriber line (ADSL): A new transport technology for delivering wideband capabilities to the residence," in *Globecom '91*, 1991.

[5] "Integrated services digital network (ISDN)-basic access interface for use on metallic loops for application on the network side of the NT (layer 1 specification)," American National Standard for Telecommunications, American National Standards Institute, Inc, Feb. 1992.

[6] J. W. Lechleider, "Coordinated transmission for two-pair digital subscriber lines," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 920–930, Aug. 1991.

[7] K. J. Kerpez, "The range of baseband ADSL's as a function of bit rate," in *IEEE Globecom '92*, vol. 1, (Orlando, FL), Dec. 1992.

[8] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* New York: John Wiley, 1991.

[9] I. Kalet and S. Shamai, "On the capacity of a twisted-wire pair: Gaussian model," *IEEE Transactions on Communications*, vol. 38, pp. 379–383, Mar. 1990.

[10] J. Lechleider, "The capacity of NEXT-impaired subscriber loops," in *IEEE Globecom '90*, vol. 2, pp. 1161–1165, Dec. 1990.

[11] K. J. Kerpez, "Near-end crosstalk is almost gaussian," *IEEE Trans. on Commun.*, vol. 41, pp. 670–672, Jan. 1993.

[12] A. Sendonaris, V. Veeravalli, and B. Aazhang, "Joint signaling strategies for maximizing the capacity of twisted pair loops," Tech. Rep. 9508, Rice University, Houston, Texas, July 1995.

[13] I. Kalet, "The multitone channel," *IEEE Transactions on Communications*, vol. 37, Feb. 1989.

[14] J. A. C. Bingham, "Multicarrier modulation for data transmission: An idea whose time has come," *IEEE Communications Magazine*, May 1990.

[15] A. Khandani, P. Kabal, and H. Leib, "Combined coding and shaping over a multitone channel," in *IEEE Globecom '91*, (Phoenix, AZ), Dec. 1991.

[16] M. Tzannes, M. Tzannes, and H. Resnikoff, "The DWMT : A multicarrier transceiver for ADSL using M-band wavelet transforms," tech. rep., T1E1.4, Mar. 1993.

[17] J. Chow, J. Tu, and J.M.Cioffi, "A discrete multitone transceiver for HDSL applications," *IEEE Journal on Selected Areas in Comm.*, vol. 9, Aug. 1991.

[18] S. Shamai, "On the capacity of a twisted-wire pair: Peak-power constraint," *IEEE Transactions on Communications*, vol. 38, pp. 368–378, Mar. 1990.

Tight Upper Bounds on the Bit Error Probability for Diversity Reception in Slow Rayleigh Fading

Saud A. Al-Semari and Thomas E. Fuja
Department of Electrical Engineering
Institute for Systems Research
University of Maryland
College Park MD 20742

Abstract

This paper derives upper bounds on the bit error probability of a trellis-coded modulation (TCM) scheme assuming diversity reception over a channel with slow, flat Rayleigh fading. Three different diversity combining techniques are investigated: maximal ratio combining, equal gain combining, and selection combining. In each case a new bound on the pairwise error probability is derived, resulting in a bound on the bit error probability through an application of the transfer-function method. The new bounds are significantly tighter than the Chernoff bounds; they permit ready comparisons between the different combining techniques and different orders of diversity.

1 Introduction

Signal fading is the primary source of performance degradation in mobile radio systems. Fading imposes extreme variations on the envelope of the transmitted signal; in land mobile radio systems, the signal envelope is often assumed to fade with a Rayleigh distribution.

Diversity combining is a well-known and effective method for improving the performance of digital communication systems over fading channels [1, 2]. The basic principle of M-fold diversity is to use M independent channels so that the probability of a "deep fade" on *all* channels is low. (These independent channels can be created in a number of ways, including frequency, time, and/or polarization division multiplexing; if multiple antennas are used to receive multiple independent versions of the signal, the approach is called *spatial* diversity.) A combining circuit is used to form a single resultant signal from the M "branch" signals. There are (at least) three different methods for combining.

- The optimal combining scheme is called *maximal ratio combining* (MRC) [2]. In such a scheme the M received K-dimensional signals are treated as a single MK-dimensional signal; the resultant SNR at the output of the combiner is the sum of the SNR's of the M branches.

- In *equal gain combining* (EGC) the resultant signal is simply the sum of the signals from the M branches.

- In *selection combining* (SC) the resultant signal is the one with highest SNR among the M received signals; in practice, the signal with the strongest received signal – i.e., signal plus noise – is selected.

Error probability expressions for uncoded systems with different combining schemes in Rayleigh fading are presented in [3, 4].

Bandwidth efficient coding, such as trellis-coded modulation (TCM) [5], also provides a form of diversity – *time diversity.* The use of TCM for mitigating the effects of fading has received considerable attention [5, 6]. The performance of a code over the Rayleigh-distributed fading channel is strongly dependent on its *minimum time diversity* – i.e., the minimum Hamming distance (in channel symbols) between two valid transmitted sequences; conversely, the performance depends very weakly on the minimum squared Euclidean distance of the code, the most important criterion for AWGN channels.

The performance of TCM schemes may be evaluated using the pairwise error probability and the transfer function approach [5]. Upper bounds on the bit error probability for different combining schemes were presented in [7, 8]. These expressions use the Chernoff bound to establish an upper limit on the pairwise error probability and are loose. A tighter upper bound on the bit error probability for maximal ratio combining (only) has recently been developed by Ventura-Traveset, Caire, Biglieri, and Taricco [9]; the upper bound in [9] requires numerical evaluation of the pairwise error probability and the use of a truncated transfer function.

In this paper, the use of trellis-coded modulation with diversity receivers is investigated. Tight upper bounds on the pairwise error probability assuming flat, slow Rayleigh fading with diversity reception and the three combining schemes above are derived. The same system configurations in [7, 8] are used; the new bounds are shown to be tighter than those presented in [7, 8].

2 System Model

The system can be described as follows. Suppose the complex signal x_i is transmitted at time i and the M corresponding signals $\underline{y}_i = \{y_{i,1}, y_{i,2}, \ldots, y_{i,M}\}$ are received; i.e.,

$$\begin{aligned} y_{i,1} &= a_{i,1}x_i + n_{i,1} \\ y_{i,2} &= a_{i,2}x_i + n_{i,2} \\ &\cdot \\ &\cdot \\ &\cdot \\ y_{i,M} &= a_{i,M}x_i + n_{i,M} \end{aligned}$$

where $\underline{a}_i = \{a_{i,1}, a_{i,2}, \ldots, a_{i,M}\}$ are the Rayleigh-distributed fading amplitudes and $\underline{n}_i = \{n_{i,1}, n_{i,2}, \ldots, n_{i,M}\}$ are noise samples, assuming two-dimensional additive white Gaussian noise with one-sided power spectral density N_0.

We assume independent branches and full interleaving – i.e., the fading values $\{a_{i,j}\}$ are independent and identically distributed with $E[a_{i,j}^2] = 1$.

The transmitter produces a sequence of signals $\mathbf{x}_N = \{x_1, x_2, \ldots, x_N\}$. At the receiver, the sequence of received M-tuples $\mathbf{y}_N = \{\underline{y}_1, \underline{y}_2, \ldots, \underline{y}_N\}$ and the channel fade amplitudes $\mathbf{a}_N = \{\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_N\}$ are the inputs to the TCM decoder, which performs maximum likelihood (ML) decoding assuming ideal channel state information. The decoding metric is

$$m(\mathbf{x}_N, \mathbf{y}_N; \mathbf{a}_N) = \sum_{i=1}^{N} m(x_i, \underline{y}_i; \underline{a}_i).$$

Here the symbol metric $m(x_i, \underline{y}_i; \underline{a}_i)$ depends on which form of signal combining is used.

- For maximal ratio combining,

$$m(x_i, \underline{y}_i; \underline{a}_i) = -\sum_{l=1}^{M} |y_{i,l} - a_{i,l}x_i|^2.$$

- For equal gain combining,

$$m(x_i, \underline{y}_i; \underline{a}_i) = -\left|\sum_{l=1}^{M} (y_{i,l} - a_{i,l}x_i)\right|^2.$$

- For selection combining,

$$m(x_i, \underline{y}_i; \underline{a}_i) = -|y_{i,j^\star} - a_{i,j^\star}x_i|^2$$

where

$$j^\star = \arg\max\{a_{i,j} : j = 1, 2, \ldots, M\}.$$

3 Pairwise Error Probability

The *pairwise error probability* of the code sequences $\mathbf{x}_N$ and $\hat{\mathbf{x}}_N$ – denoted $P(\mathbf{x}_N \to \hat{\mathbf{x}}_N)$ – is the probability that the decoder selects $\hat{\mathbf{x}}_N$ as its estimate of the transmitted sequence, given that $\mathbf{x}_N$ was, in fact, transmitted; this happens if and only if the metric associated with $\hat{\mathbf{x}}_n$ exceeds that of $\mathbf{x}_N$.

In this section, tight upper bounds on the pairwise error probability are derived for the three combining schemes.. Moreover, these pairwise error probability expressions are expressed in product form, thereby permitting the transfer function technique for bounding the bit error probability.

3.1 Maximal Ratio Combining

The conditional pairwise error probability can be expressed as

$$\begin{aligned} P(\mathbf{x}_N \to \hat{\mathbf{x}}_N|\mathbf{a}_N) &= P(m(\hat{\mathbf{x}}_N, \mathbf{y}_N; \mathbf{a}_N) - m(\mathbf{x}_N, \mathbf{y}_N; \mathbf{a}_N) \geq 0|\mathbf{a}_N) \\ &= P(\sum_{i=1}^{N}\sum_{l=1}^{M}(|y_{i,l} - a_{i,l}x_i|^2 - |y_{i,l} - a_{i,l}\hat{x}_i|^2) \geq 0|\mathbf{a}_N) \end{aligned}$$

which can be simplified to

$$P(\mathbf{x}_N \to \hat{\mathbf{x}}_N|\mathbf{a}_N) = \frac{1}{2}\mathrm{erfc}\left(\sqrt{\sum_{i=1}^{N}\gamma_i d_i}\right)$$

where $\mathrm{erfc}(x) = \frac{2}{\sqrt{\pi}}\int_x^\infty e^{-t^2}dt$, $d_i = |x_i - \hat{x}_i|^2/4N_0$, and $\gamma_i = \sum_{l=1}^M a_{i,l}^2$. Since the $a_{i,l}$'s are i.i.d. Rayleigh distributed, their squares are i.i.d. exponentially distributed, and so γ_i has an M-Erlang distribution with parameter one – i.e., its pdf is

$$p(\gamma_i) = \frac{1}{(M-1)!}\gamma_i^{(M-1)}e^{-\gamma_i}.$$

The unconditional pairwise error probability is thus

$$P(\mathbf{x}_N \to \hat{\mathbf{x}}_N) = \frac{1}{2}\int_0^\infty \cdots \int_0^\infty \mathrm{erfc}\left(\sqrt{\sum_{i=1}^N \gamma_i d_i}\right) p(\gamma_1)\cdots p(\gamma_N)d\gamma_1\cdots d\gamma_N.$$

Define

$$\delta_i = \frac{d_i}{1+d_i} \qquad \text{and} \qquad \omega_i = \gamma_i(1+d_i).$$

Then the unconditional pairwise error probability can be represented as

$$\begin{aligned} P(\mathbf{x}_N \to \hat{\mathbf{x}}_N) &= \frac{1}{2}\prod_{i\in\eta}\frac{1}{(1+d_i)^M}\int_0^\infty \cdots \int_0^\infty \mathrm{erfc}\left(\sqrt{\sum_{i=1}^N \delta_i\omega_i}\right) \\ &\quad \times \exp[\sum_{i=1}^N \delta_i\omega_i]\, p(\omega_1)\cdots p(\omega_N)d\omega_1\cdots d\omega_N \end{aligned}$$

where $\eta = \{i : x_i \neq \hat{x}_i\}$, $L_\eta = |\eta|$, and $p(\omega_i)$ is the M-Erlang distribution of ω_i with parameter one. But,

$$\sum_{i\in\eta}\delta_i\omega_i \geq \delta_m \sum_{i\in\eta}\omega_i$$

where $\delta_m = \min\{\delta_i : i \in \eta\}$. Since $\mathrm{erfc}(x)e^{x^2}$ is monotonically decreasing for $x \geq 0$, the pairwise error probability can be upper bounded by

$$P(\mathbf{x}_N \to \hat{\mathbf{x}}_N) \leq \frac{1}{2}\prod_{i\in\eta}\frac{1}{(1+d_i)^M}\int_0^\infty \mathrm{erfc}\left(\sqrt{\delta_m\Omega}\right)e^{\delta_m\Omega}f(\Omega)d\Omega$$

where $\Omega = \sum_{i\in\eta}\omega_i$, and $f(\Omega)$ is the (ML_η)-Erlang distribution with parameter one. To evaluate the integral, we use the following identity [6]

$$\frac{1}{2(K-1)!}\int_0^\infty \mathrm{erfc}(\sqrt{xy})e^{-y(1-x)}y^{(K-1)}dy = \frac{1}{2^{2K}}\sum_{j=1}^K \binom{2K-j-1}{K-1}\left(\frac{2}{1+\sqrt{x}}\right)^j,$$

which is valid for $x < 1$. Integration yields

$$P(\mathbf{x}_N \to \hat{\mathbf{x}}_N) \leq \left[\frac{1}{2^{2ML_\eta}}\sum_{j=1}^{ML_\eta}\binom{2ML_\eta - j - 1}{ML_\eta - 1}\left(\frac{2}{1+\sqrt{\delta_m}}\right)^j\right] \times \prod_{i\in\eta}\frac{1}{(1+d_i)^M}.$$

Let L be the minimum time diversity of the code – i.e., the minimum Hamming distance, in signal symbols, between two code sequences. Then $L \leq L_\eta$ and we can further upper bound the pairwise error probability by

$$P(\mathbf{x}_N \to \hat{\mathbf{x}}_N) \leq \left[\frac{1}{2^{2ML}}\sum_{j=1}^{ML}\binom{2ML - j - 1}{ML - 1}\left(\frac{2}{1+\sqrt{\delta_m}}\right)^j\right] \times \prod_{i\in\eta}\frac{1}{(1+d_i)^M}$$

Note that the upper bound in [6] is a special case of this bound ($M = 1$).

3.2 Equal Gain Combining

For equal gain combining, the conditional pairwise error probability can be expressed as

$$\begin{aligned} P(\mathbf{x}_N \to \hat{\mathbf{x}}_N | \mathbf{a}_N) &= P(m(\hat{\mathbf{x}}_N, \mathbf{y}_N; \mathbf{a}_N) - m(\mathbf{x}_N, \mathbf{y}_N; \mathbf{a}_N) \geq 0 | \mathbf{a}_N) \\ &= P(\sum_{i=1}^{N} (|\sum_{l=1}^{M} (y_{i,l} - a_{i,l} x_i)|^2 - |\sum_{l=1}^{M} (y_{i,l} - a_{i,l} \hat{x}_i)|^2) \geq 0 | \mathbf{a}_N) \\ &= \frac{1}{2} \mathrm{erfc} \left(\sqrt{\sum_{i=1}^{N} \mu_i \tilde{d}_i / b_o} \right). \end{aligned}$$

where $b_o = [(2M-1)!!]^{1/M}$, $\tilde{d}_i = b_o |x_i - \hat{x}_i|^2 / 4MN_0$ and $\mu_i = (\sum_{l=1}^{M} a_{i,l})^2$. However, there is no closed form for the pdf of the sum of M Rayleigh distributed random variables for $M > 2$, so an approximation will be used. The approximation is based on the small argument approximation [1, 2] and is given by

$$f(\mu_i) = \frac{\mu_i^{(M-1)} \exp(-\frac{\mu_i}{b_o})}{b_o^M (M-1)!}$$

for $\mu_i \geq 0$. (Beaulieu [11] showed that this expression is very accurate for $M \leq 8$.) Defining $\gamma_i = \mu / b_o$, then the pdf of γ_i is

$$f(\gamma_i) = \frac{\gamma_i^{(M-1)} \exp(-\gamma_i)}{(M-1)!},$$

again for $\gamma_i \geq 0$. Then the unconditional pairwise error probability can be expressed as

$$P(\mathbf{x}_N \to \hat{\mathbf{x}}_N) = \frac{1}{2} \int_0^\infty \cdots \int_0^\infty \mathrm{erfc} \left(\sqrt{\sum_{i=1}^{N} \gamma_i \tilde{d}_i} \right) \times f(\gamma_1) \cdots f(\gamma_N) d\gamma_1 \cdots d\gamma_N$$

Similar to the maximal ratio combining case, define

$$\tilde{\delta}_i = \frac{\tilde{d}_i}{1 + \tilde{d}_i} \quad \text{and} \quad \tilde{\omega}_i = \gamma_i (1 + \tilde{d}_i)$$

and note that

$$\sum_{i \in \eta} \tilde{\delta}_i \tilde{\omega}_i \geq \delta_e \sum_{i \in \eta} \tilde{\omega}_i$$

where $\delta_e = \min\{\tilde{\delta}_i : i \in \eta\}$. Then the pairwise error probability can be bounded by

$$P(\mathbf{x}_N \to \hat{\mathbf{x}}_N) \leq \left[\frac{1}{2^{2ML}} \sum_{j=1}^{ML} \binom{2ML - j - 1}{ML - 1} \left(\frac{2}{1 + \sqrt{\delta_e}} \right)^j \right] \times \prod_{i \in \eta} \frac{1}{(1 + \tilde{d}_i)^M}$$

3.3 Selection Combining

With selection combining, the conditional pairwise error probability is given by

$$\begin{aligned} P(\mathbf{x}_N \to \hat{\mathbf{x}}_N | \mathbf{a}_N) &= P(m(\hat{\mathbf{x}}_N, \mathbf{y}_N; \mathbf{a}_N) - m(\mathbf{x}_N, \mathbf{y}_N; \mathbf{a}_N) \geq 0 | \mathbf{a}_N) \\ &= P(\sum_{i=1}^{N} (|y_{i,l} - a_{i,j^\star} x_i|^2 - |y_{i,l} - a_{i,j^\star} \hat{x}_i|^2) \geq 0 | \mathbf{a}_N) \\ &= \frac{1}{2} \mathrm{erfc} \left(\sqrt{\sum_{i=1}^{N} \nu_i d_i} \right). \end{aligned}$$

where $d_i = |x_i - \hat{x}_i|^2/4N_0$, and the pdf of ν_i is

$$\begin{aligned} f(\nu_i) &= M(1-e^{-\nu_i})^{(M-1)}e^{-\nu_i} \\ &= \sum_{k_i=1}^{M} M(-1)^{k_i+1}\begin{pmatrix} M-1 \\ k_i-1 \end{pmatrix} e^{-k_i\nu_i} \end{aligned}$$

The unconditional pairwise error probability can be expressed as

$$\begin{aligned} P(\mathbf{x}_N \to \hat{\mathbf{x}}_N) &= \frac{1}{2}\sum_{k_1=1}^{M}\cdots\sum_{K_N=1}^{M}\prod_{i=1}^{N}\left\{M(-1)^{k_i+1}\begin{pmatrix} M-1 \\ k_i-1 \end{pmatrix}\right\} \\ &\quad \cdot \int_0^\infty \cdots \int_0^\infty \operatorname{erfc}\left(\sqrt{\sum_{i=1}^{N}\nu_i d_i}\right) e^{-k_1\nu_1}\cdots e^{-k_N\nu_N}\,d\nu_1\cdots d\nu_N \end{aligned}$$

Define

$$\delta_{i,k_i} = \frac{d_i}{k_i + d_i} \qquad \text{and} \qquad \omega_{i,k_i} = \nu_i(k_i + d_i).$$

and observe that

$$\sum_{i\in\eta}\delta_{i,k_i}\omega_{i,k_i} \geq \delta_s \sum_{i\in\eta}\omega_{i,k_i}$$

where

$$\delta_s = \min\{\delta_{i,k_i} : i \in \eta, k_i \in \{1,\ldots M\}\} = \frac{\min\{d_i\}}{M + \min\{d_i\}}$$

This yields the following bound on the pairwise error probability assuming selection combining:

$$\begin{aligned} P(\mathbf{x}_N \to \hat{\mathbf{x}}_N) &\leq \left[\frac{1}{2^{2L}}\sum_{j=1}^{L}\begin{pmatrix} 2L-j-1 \\ L-1 \end{pmatrix}\frac{2^j}{(1+\sqrt{\delta_s})^j}\right] \\ &\quad \prod_{i\in\eta}\left[\sum_{k=1}^{M} M(-1)^{k+1}\begin{pmatrix} M-1 \\ k-1 \end{pmatrix}\frac{1}{(k+d_i)}\right] \end{aligned}$$

4 Bit Error Probability of Trellis Coded Systems

The average bit error probability, P_b, of trellis coded systems is bounded [5] by

$$P_b \leq \sum_{n=1}^{\infty}\sum_{\mathbf{x}_n}\sum_{\mathbf{x}_n\neq\hat{\mathbf{x}}_n} N_n(\mathbf{x}_n, \hat{\mathbf{x}}_n)\Pr(\mathbf{x}_n)\Pr(\mathbf{x}_n \to \hat{\mathbf{x}}_n)$$

where $\Pr(\mathbf{x}_n)$ is the a priori probability of transmitting $\mathbf{x}_n$, $\Pr(\mathbf{x}_n \to \hat{\mathbf{x}}_n)$ is the pairwise error probability of the sequences $\mathbf{x}_n$ and $\hat{\mathbf{x}}_n$, and $N_n(\mathbf{x}_n, \hat{\mathbf{x}}_n)$ is the number of bit errors occurring when $\mathbf{x}_n$ is transmitted and $\hat{\mathbf{x}}_n$ is the decoder's estimate. Note that [5]

$$T(\bar{D}, I) = \sum_{n=1}^{\infty}\sum_{\mathbf{x}_n}\sum_{\mathbf{x}_n\neq\hat{\mathbf{x}}_n}\Pr(\mathbf{x}_n)I^{N_n(\mathbf{x}_n,\hat{\mathbf{x}}_n)}\prod_{i=1}^{N}W(x_i, \hat{x}_i)$$

where $T(\bar{D}, I)$ is the transfer function of the code with D averaged over the fading distribution and $W(x_i, \hat{x}_i)$ is the error weight profile at time i. Therefore, bit error probability can be expressed as follows.

- Maximal ratio combiner (MRC):

$$P_b \leq \frac{1}{k}\left[\frac{1}{2^{2ML}}\sum_{j=1}^{ML}\binom{2ML-j-1}{ML-1}\left(\frac{2}{1+\sqrt{\delta_{\min}}}\right)^j\right]\frac{\partial T(\bar{D},I)}{\partial I}\,|_{I=1,D=e^{-E_s/4N_0}}$$

- Equal gain combining (EGC):

$$P_b \leq \frac{1}{k}\left[\frac{1}{2^{2ML}}\sum_{j=1}^{ML}\binom{2ML-j-1}{ML-1}\left(\frac{2}{1+\sqrt{\delta_e}}\right)^j\right]\frac{\partial T(\bar{D},I)}{\partial I}\,|_{I=1,D=e^{-E_s/4N_0}}$$

- Selection combining (SC):

$$P_b \leq \frac{1}{k}\left[\frac{1}{2^{2L}}\sum_{j=1}^{L}\binom{2L-j-1}{L-1}\left(\frac{2}{1+\sqrt{\delta_s}}\right)^j\right]\frac{\partial T(\bar{D},I)}{\partial I}\,|_{I=1,D=e^{-E_s/4N_0}}$$

where

$$\bar{D}\,|_{D=e^{-E_s/4N_0}} = \begin{cases} \frac{1}{(1+\frac{E_s}{4N_0})^M} & \text{for MRC} \\ \frac{1}{(1+\frac{b_0 E_s}{4MN_0})^M} & \text{for EGC} \\ \sum_{k=1}^{M} M(-1)^{k+1}\binom{M-1}{k-1}\frac{1}{k+\frac{E_s}{4N_0}} & \text{for SC} \end{cases}$$

This bound is no more complex than the Chernoff bound. Moreover, the bracketed term to the left of the transfer function is at most one, so this bound is at least as tight as the Chernoff bound. The following examples will show that it is, in fact, tighter.

4.1 Examples

An 8-state I-Q TCM code employing 16-QAM is used as an example. I-Q TCM codes are trellis codes in which the in-phase and quadrature components of the transmitted signal are encoded independently; Al-Semari and Fuja [12] have shown that this approach yields better performance over Rayleigh fading channels than codes designed using the "traditional" approach. This particular code has a minimum time diversity of $L = 4$, and its performance is superior to that of the comparable 8-state conventional TCM code employing 8-PSK in Rayleigh fading. (See [12] for details.)

Figure 1 compares the newly derived bound for dual diversity (M=2) and maximal ratio combining with the Chernoff bound for the same code; the new bound is slightly more than 1 dB tighter than the Chernoff bound at a bit error rate of 10^{-5}.

The performance of the 8-state 16-QAM code with MRC and different orders of diversity is shown in Figure 2. It is clear that the largest incremental gain is obtained in going from single to double diversity; the coding gains diminish as the order of diversity increases. Similar results are displayed for equal gain and selection combining in Figures 3 and 4; for each combining scheme, incremental coding gains diminish with increasing diversity. Comparisons between the three combining schemes show that MRC achieves the best performance; however, EGC error performance is within 1 dB of MRC. As M increases, the difference between the three schemes increases. For example, with dual diversity ($M = 2$), maximal ratio combining has a gain of approximately 1.2 dB over selection combining at a BER of 10^{-5}; however, with four-fold diversity ($M = 4$), the

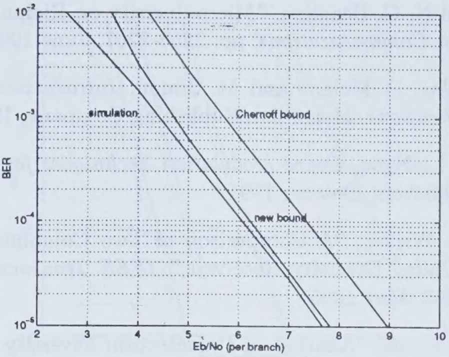


Figure 1: A comparison between the new bound and the Chernoff bound for the 16-QAM I-Q TCM 8-state code with maximal ratio combining and double diversity.

gain is more than 3 dB. Finally, it is noted that the upper bound is very tight and gives very accurate values for the bit error rate, especially at bit error rate less than 10^{-3}.

Figure 5 compares the effects of diversity on the 8-state 16-QAM I-Q code and a "conventional" 8-state 8-PSK trellis code optimized for time diversity. Double diversity ($M = 2$) and maximal ratio combining are used. It is observed that the gain enjoyed by I-Q TCM in single-channel applications is evident with double diversity as well; the I-Q code outperforms the 8-PSK code by about 1.5 dB at a BER=10^{-5}.

5 Conclusions

In this paper, new upper bounds on the pairwise error probability of coherent systems with maximal ratio, equal gain, and selection combining schemes are derived. These upper bounds are used to evaluate the BER of a variety of system configurations. Simulations show that the new bounds are quite tight; moreover, they show that there is a "diminishing returns" effect as diversity is added, regardless of the combining technique. However, the relative advantage of maximal ratio combining over equal gain and selection combining actually *increases* with the diversity order. Finally, the bounds demonstrate that, when I-Q TCM 16-QAM is compared with 8-PSK trellis codes optimized for the fading channel with dual diversity, the I-Q TCM code enjoys a coding gain of approximately 1.5 dB at a BER of 10^{-5}.

References

[1] M. Schwartz W. Bennett and S. Stein, *Communication Systems and Techniques,* McGraw-Hill, New York, 1966.

[2] W. C. Jakes, Jr., *Microwave Mobile Communications,* John Wiley and Sons, New York, 1974.

[3] W. Lindsey, "Error probabilities for Rician fading multichannel reception of binary and N-ary signals," *IEEE Trans. on Information Theory*, pp. 339-350, October 1964.

[4] A. A. Abu-Dayya and N. C. Beaulieu, "Microdiversity on Rician fading channels," *IEEE Transactions on Communications,* pp. 2258-2267, June 1994.

[5] E. Biglieri, D. Divsalar, P. Mclane and M. Simon, *Introduction to Trellis-Coded Modulation with Applications,* Macmillan Publishing Company, 1991.

[6] S. H. Jamali, and T. Le-Ngoc, *Coded-Modulation Techniques for Fading Channels,* Kluwer Academic Publishers, Boston, 1994.

[7] L. Rasmussen and S. Wicker, "A Comparison of Two Combining Techniques for Equal Gain, Trellis Coded Diversity Receivers," *IEEE Transactions on Vehicular Technology,* pp. 291-295, May 1995.

[8] G. Femenias and R. Agusti, "Analysis of Predetection Diversity TCM-MPSK and Postdetection Diversity TCM-MDPSK Systems on a Rayleigh Fading Channel," *IEEE Transaction on Communications,* pp. 374-385, Feb./Mar./Apr. 1995.

[9] J. Ventura-Traveset, G. Caire, E. Biglieri and G. Taricco, "Impact of Diversity Reception on Fading Channels with Coded Modulation. Part I: Coherent Detection," preprint.

[10] C. Schlegel, and D. J. Costello, Jr. "Bandwidth Efficient Coding for Fading Channels: Code Construction and Performance Analysis", *IEEE Journal on Selected Areas in Communications,* pp. 1356-1368, December 1989,

[11] N. C. Beaulieu, "An infinite series for the computation of the complementary probability distribution function of a sum of independent random variables and its application to the sum of Rayleigh random variables," *IEEE Transactions on Communications,* pp. 1463-1474, September 1990.

[12] S. A. Al-Semari, and T. E. Fuja, "I-Q TCM: Reliable Communication Over the Rayleigh Fading Channel Close to the Cutoff Rate," submitted to *IEEE Transactions on Information Theory.*

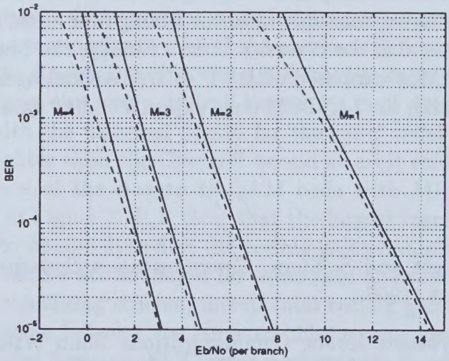


Figure 2: BER of 16-QAM I-Q TCM 8-state code with maximal ratio combining and different diversity orders. solid(bound), dashed(simulation).

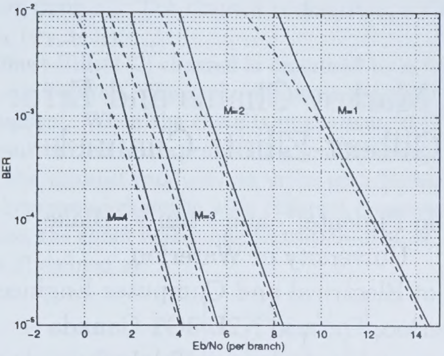


Figure 3: BER of 16-QAM I-Q TCM 8-state code with equal gain combining and different diversity orders. solid(bound), dashed(simulation).

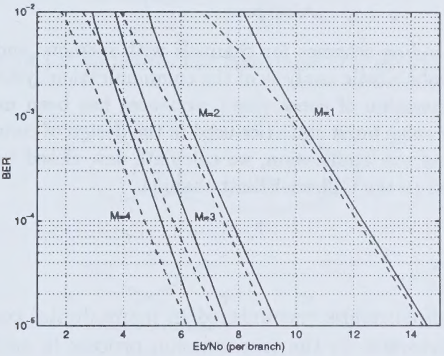


Figure 4: BER of 16-QAM I-Q TCM 8-state code with selection combining and different diversity orders. solid(bound), dashed(simulation).

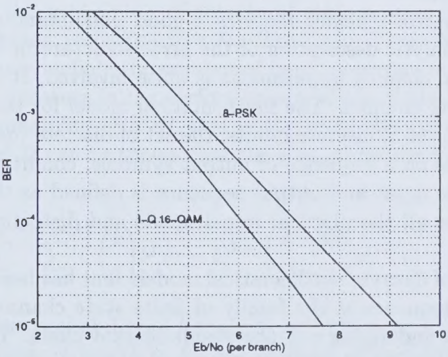


Figure 5: Simulated BER of 16-QAM I-Q TCM 8-state code and the 8-PSK TCM 8-state code with MRC and double diversity.

Enumeration of Markov Chains and Error Statistics for Gilbert-Elliott Channels

Cecilio Pimentel* and Ian F. Blake
University of Waterloo
Department of Electrical and Computer Engineering
Waterloo, Ontario N2L 3G1 Canada
E. mail: cpimente@claude.uwaterloo.ca, ifblake@claude.uwaterloo.ca

Abstract

In order to design coding schemes for channels with memory and to evaluate their performance, a probabilistic analysis of the communication system is needed. The theory of enumeration of constrained sequences has been used to derive analytic expressions for some burst statistics used in the design of coded system for finite state channels. As one application, we calculate new closed form solutions for two statistics related to the Gilbert-Elliott channel.

1 Introduction

The effect of the channel impairments encountered in many digital communication systems tends to introduce distortion in the transmission process in such a way that the errors are grouped together in bursts. The statistical dependence in the occurrence of errors introduces memory into the error process. The achievement of high reliability in data transmission over channels with memory will depend on the design and implementation of an effective coding scheme.

The design of error control schemes requires a good understanding of the channel behavior. However, the analytic description of the error structure in terms of the characteristics of the individual channel impairments is quite involved. It may be more convenient to specify a discrete channel. The mathematical model for the discrete channel has three constituents: the set of discrete inputs, the set of discrete (quantized) outputs, and the set of probabilities on a sequence of output symbols, conditional on the input. The difference between the input and output sequence is defined as the *error sequence.* This sequence incorporates all the physical impairments and distortions present in the channel.

One important family of discrete mathematical models that has been extensively used to characterize the error sequence is the family of finite state channels (FSC) [1]. An FSC is characterized by an underlying non-observable Markov chain. The output symbol of the FSC at the k^{th} time interval, y_k, is a function of the input symbol x_k and the

---

*Supported by the Brazilian Council for Scientific and Technological Research (CNPq).

state of the Markov chain s_k. The channel is described statistically by the conditional probability $p(y_k, s_k \mid x_k, s_{k-1})$.

The binary Gilbert-Elliott [2] channel is described by a two-state Markov chain with state space $\{0, 1\}$, see Figure 1. The error process is generated according to the following probabilistic mechanism. When the chain is in the state 0 (the good state) the channel corrupts the transmitted bit with probability g. Otherwise, when it is in the state 1 (the burst state), the channel produces an error with higher probability b. Each state represents binary symmetric channels with different crossover probabilities. After every channel transmission the chain makes one transition of state according to the transition probability matrix P defined as:

$$P = \begin{bmatrix} 1-Q & Q \\ q & 1-q \end{bmatrix}. \tag{1}$$

We assume that the distribution of the initial state is the stationary distribution Π

$$\Pi = [\pi_0, \pi_1]^T = \left[\frac{q}{Q+q}, \frac{Q}{Q+q}\right]^T, \tag{2}$$

where the superscript $[\cdot]^T$ indicates the transpose of a matrix.

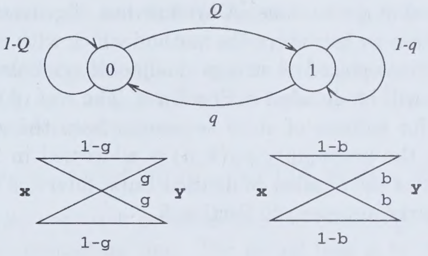


Figure 1: The Gilbert-Elliott channel.

The analysis of a communication system operating over finite state channels includes the calculation of the probability of subsets of state sequences, where each sequence is statistically distributed according to a Markov chain. Some statistics of interest for the encoder designer may include the number of burst occurrences, the average burst length, etc. An important measure in the performance analysis of block codes and interleavers is the error weight probability $p(m, n)$, that is, the probability of exactly m errors occurring in a block of n bits. It can be shown that the $p(m, n)$ for the Gilbert-Elliott channel may be expressed as [3]:

$$p(m,n) = \sum_{k=0}^{n} \sum_{j=0}^{min(m,k)} \binom{k}{j} b^j (1-b)^{k-j} \binom{n-k}{m-j} g^{m-j}(1-g)^{n-k-m+j} \, p_B(k,n), \tag{3}$$

where $p_B(k, n)$ is the probability of k transitions from state 1 (the burst state) occurring in a sequence of states of length n. So one problem of particular interest in the analysis of coded systems operating over the Gilbert-Elliott channel is the calculation of $p_B(k, n)$.

In Reference [3], Yee and Weldon employed combinatorial identities to derive an explicit expression for $p_B(k, n)$. Their approach has three main steps. First, the set of state sequences is partitioned into subsets having sequences with the same probability. Then, the number of sequences in each subset is determined. The final result follows by combining properly the probabilities of each subset.

The purpose of this paper is to present a systematic method for treating a class of problems involving the calculation of probabilities of events generated by an arbitrary Markov chain.

In this work we follow the theory of enumeration of constrained sequences described in [4, Chapter 4], also denoted as *the pattern algebra*, to enumerate all permissible sequences of state transitions of a Markov chain with an arbitrary number of states. This powerful and general approach allows us to find a simple formula for the sum of a formal power series, the generating series, whose quantities of interest are the coefficients of particular terms. The probability of a particular set of sequences is obtained by acting on the generating series with a linear mapping. The final result follows by coefficient extraction. Such an approach allows the derivation of various burst error statistics used in the design of a coded system for the channel of interest.

As one application, we specialize the general expressions to find a new closed expression for $p_B(k,n)$. This expression is more compact than previously published expressions. Moreover, we will use the generating series to find a recurrence formula for $p_B(k,n)$, which is convenient for computation. New results involving other statistics of interest will also be given.

This work is organized in six sections. A brief review of generating series is presented in Section 2. In this section we introduce the method which with enable the enumeration of sequences with restrictions placed on strings of adjacent symbols. The enumeration of a two-state Markov chain will be detailed in Section 3. The rest of this paper is dedicated to obtain probabilities for subsets of state sequences from the generating series. The problem of determining the probability $p_B(k,n)$ is addressed in Section 4. As another example, the probability of the number of distinct burst intervals is derived in Section 5. Finally, concluding remarks are given in Section 6.

2 The Generating Series for Sequences

This section has two parts. First, we briefly review the main concepts of generating series which are relevant to this work. A comprehensive treatment on this subject can be found in Reference [4]. Next, we introduce a method which enable us to enumerate sequences over a finite alphabet subject to restrictions placed on its adjacent symbols. The theory presented in this section turns out to be useful in the analysis of finite state channels.

We now define some notation. Let N be the set of all non-negative integers, and R be the field of real numbers. Let $\underline{x} = (x_1, \ldots, x_W)$ be a set of commutative indeterminates, and let $R[[\underline{x}]]$ be the ring of all formal power series in $\underline{x}$, with coefficients in R. If $f(\underline{x}) \in R[[\underline{x}]]$, then $[x_1^{i_1} x_2^{i_2} \ldots x_W^{i_W}]\, f(\underline{x})$ denotes the coefficient of the term $x_1^{i_1} x_2^{i_2} \ldots x_W^{i_W}$ in the series $f(\underline{x})$. The $(i,j)^{th}$ entry of an $n \times n$ matrix B will be denoted by $[B]_{i,j}$, where $0 \leq i,j \leq n-1$. The identity matrix will be denoted by I.

2.1 The Generating Series in Commuting Indeterminates

Let $\mathcal{S}$ be the set of all permissible sequences over the alphabet $\mathcal{N}_A = \{0, \ldots, A-1\}$. A generic sequence will be denoted by $\sigma = (\sigma_1, \sigma_2, \ldots)$, $\sigma_i \in \mathcal{N}_A$. Let $\sigma_n = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ be a sequence of length n. Notice that in general $\mathcal{S} \subseteq \mathcal{N}_A^\star$, where $\mathcal{N}_A^\star$ is the set of all sequences of any length whose symbols σ_i are in $\mathcal{N}_A$. The type of σ is the A-tuple $\tau(\sigma) = (i_0, i_1, \ldots, i_{A-1})$, where the integer i_j is the number of occurrences of the symbol j in σ. For example, the type of the sequence $\sigma = (0,1,1,0,1,1)$ over $\mathcal{N}_2$ is $\tau(\sigma) = (2,4)$.

Let w_1, w_2,..., w_W be the set of weight functions, where $w_i : \mathcal{S} \to N$, $i = 1, \ldots, W$. Each univariate weight function w_i records designed information about the sequence σ. We say that $w_i(\sigma)$ is the weight of σ with respect to w_i. These univariate weight functions are grouped together by defining the following composite weight function:

$$w_1 \otimes w_2 \otimes \ldots \otimes w_W : \mathcal{S} \to N^W. \tag{4}$$

The notation N^W in (4) means a W-dimensional tuple with elements in N. The left hand side of Equation (4) is merely a notational device. This is to be interpreted as the enumeration of the set $\mathcal{S}$ with respect to a variety of characteristics. This leads to the definition of a multivariate generating series.

The ordinary multivariate generating series for $\mathcal{S}$ with respect to weight function $w_1 \otimes \ldots \otimes w_W$, denoted by $[(\mathcal{S}, w_1 \otimes \ldots \otimes w_W)]$, in the indeterminates $\underline{x} = (x_1, \ldots, x_W)$ is defined by:

$$[(\mathcal{S}, w_1 \otimes \ldots \otimes w_W)] = \sum_{\sigma \in \mathcal{S}} x_1^{w_1(\sigma)}\, x_2^{w_2(\sigma)} \ldots x_W^{w_W(\sigma)}. \tag{5}$$

The generating series in (5) is to be regarded as a formal power series in $R[[\underline{x}]]$ with indeterminates $\underline{x}$, which retain combinatorial information on coefficients. We say that the indeterminate x_i marks w_i, and that $[(\mathcal{S}, w_1 \otimes \ldots \otimes w_W)]$ enumerates $\mathcal{S}$ with respect to $w_1 \otimes \ldots \otimes w_W$.

The general enumeration problem may now be formulated as follows: Find the number of distinct sequences $c(i_1, i_2, \ldots, i_W)$ in $\mathcal{S}$ with weight $w_1 \otimes \ldots \otimes w_W(\sigma) = (i_1, i_2, \ldots, i_W)$. It is a straightforward observation that:

$$c(i_1, i_2, \ldots, i_W) = [x_1^{i_1}\, x_2^{i_2} \ldots x_W^{i_W}][(\mathcal{S}, w_1 \otimes \ldots \otimes w_W)]. \tag{6}$$

The generating series approach provides us with a strategy for solving problems involving sequences in a commuting ring. The initial task is to define the set $\mathcal{S}$ and the composite weight function (4). The next step is to derive an expression for the generating series. The desired quantity $c(i_1, i_2, \ldots, i_W)$ will be obtained by extracting the coefficient of the appropriate power of the indeterminates.

2.2 Generating Series in Non-commuting Indeterminates

Recall that the general problem we will address in this paper is to find the probability of events associated with state sequences of a Markov chain. It is well known that the probability of a sequence is not preserved under commutation of its symbols. This prompts us to define the generating series in non-commuting indeterminates in order to keep all the information about the original sequence. The generating series for $\mathcal{S}$ is

$$F_{\mathcal{S}} = \sum_{\sigma \in \mathcal{S}} x_{\sigma_1}\, x_{\sigma_2} \ldots x_{\sigma_n}, \tag{7}$$

which is in $R \ll x_0, \ldots, x_{A-1} \gg$, the ring of all power series in the non-commuting indeterminates $x_0, \ldots, x_{A-1}$. For example, if $\mathcal{S}'$ is the subset of $\mathcal{N}_2^\star$ defined as $\mathcal{S}'$={ one 1 in a block of length 4 }, that is, $\mathcal{S}' = \{1000, 0100, 0010, 0001\}$. According to the definition of $F_{\mathcal{S}}$ in (7) the generating series for the set defined above is

$$F_{\mathcal{S}'} = x_1 x_0^3 + x_0 x_1 x_0^2 + x_0^2 x_1 x_0 + x_0^3 x_1, \qquad \in\ R \ll x_0, x_1 \gg . \tag{8}$$

The generating series $F_{\mathcal{S}}$ is obtained from $\mathcal{S}$ by replacing commas by plus signs and σ_i by $x_{\sigma i}$. The commutative image of $F_{\mathcal{S}}$ is the element of $R[[x_0, \ldots, x_{A-1}]]$ that is obtained from $F_{\mathcal{S}}$ when $x_0, \ldots, x_{A-1}$ commute. Let $\chi F_{\mathcal{S}}$ denote the commutative image of $F_{\mathcal{S}}$. Then, calling τ the weight function that determines the type

$$[(\mathcal{S}', \tau)] = \chi F_{\mathcal{S}'} = 4\, x_0\, x_1^3.$$

We will now introduce the general method for determining the generating series of all permissible sequences in $\mathcal{S}$ with various restrictions placed on substrings of adjacent symbols.

Let B be the $A \times A$ *adjacency matrix* whose entries are defined as:

$$[B]_{i,j} = \begin{cases} y_{i,j}, & \text{if } (\sigma_k = i, \sigma_{k+1} = j); \\ 0, & \text{if } (\sigma_k = i, \sigma_{k+1} = j) \text{is not allowed,} \end{cases} \tag{9}$$

where the indeterminates $y_{i,j}$ mark the occurrence of substrings of adjacent symbols $(\sigma_k = i, \sigma_{k+1} = j)$ in σ. Let $\mathcal{S}^n$ be the subset of $\mathcal{S}$ composed by all sequences in $\mathcal{S}$ whose length is n. Then, the generating series for the set $\mathcal{S}^n$ with respect to the occurrence of symbols and the transition between adjacent symbols may be expressed as:

$$F_{\mathcal{S}^n} = \sum_{\sigma_n \in \mathcal{N}_A^n} x_{\sigma_1}\, [B]_{\sigma_1,\sigma_2}\, x_{\sigma_2} \, \ldots x_{\sigma_{n-1}}\, [B]_{\sigma_{n-1},\sigma_n}\, x_{\sigma_n}, \tag{10}$$

Let X be the diagonal matrix of indeterminates:

$$X = \operatorname{diag}(x_0, \ldots, x_{A-1}). \tag{11}$$

Then, it can be shown that

$$\begin{aligned} F_{\mathcal{S}^n} &= \sum_{0 \le i,j \le A-1} [(XB)^{n-1}\, X]_{i,j} \\ &= \operatorname{trace}((X\, B)^{n-1}\, X\, J), \end{aligned} \tag{12}$$

where J is an $A \times A$ matrix whose all entries are one. We can now write $F_{\mathcal{S}}$ as:

$$F_{\mathcal{S}} = \sum_{n=1}^{\infty} F_{\mathcal{S}^n} = \operatorname{trace}((I - X\, B)^{-1}\, X\, J) \qquad \in R \ll x_0, \ldots, x_{A-1} \gg. \tag{13}$$

If the weight function w is defined as

$$w = \tau \otimes w_{0,0} \otimes \ldots \otimes w_{A-1,A-1}, \tag{14}$$

where $\tau(\sigma)$ is the type of σ and each $w_{i,j}(\sigma)$ counts the number of occurrences of substrings of adjacent symbols $(\sigma_k = i, \sigma_{k+1} = j)$ in σ, $i, j \in 0, \ldots, A-1$, we can write $[(\mathcal{S}, w)]$ very compactly as

$$[(\mathcal{S}, w)] = \chi \operatorname{trace}((I - X\, B)^{-1}\, X\, J) \qquad \in R[[x_0, \ldots, x_{A-1}]]. \tag{15}$$

For the remainder of this section we consider examples of the use of Equation (15) in the enumeration of sequences generated by Markov chains. The generating series associated with a given problem may be obtained by a transformation of (15). When the set $\mathcal{S}$ has restriction on its initial or final symbol, the summation in (12) must be carried out only for specific values of (i, j). The process of transformation may also involve the insertion of indeterminates to record the appropriate combinatorial information. These transformations are better illustrated with an example. In the next section, we will develop an expression for the generating series of a two-state Markov chain with respect to the number of transitions between states.

3 Enumeration of a Two-state Markov Chain

Consider a two-state Markov chain with state space $\{0, 1\}$. According to the notation of the Gilbert-Elliott channel, state 0 represents the good state, and state 1 represents the burst state. Let $\mathcal{S}$ be the set of all sequences of states over $\mathcal{N}_2 = \{0, 1\}$, i.e., $\mathcal{S} = \mathcal{N}_2^\star$. Let $w_{0,0}$, $w_{0,1}$, $w_{1,0}$, $w_{1,1}$ be weight functions $w_{i,j} : \{0,1\}^\star \to N, i, j = 0, \ldots, 1$, such that $w_{i,j}(\sigma)$ is the number of occurrences of the substring (i, j) in σ.

In this section we wish to determine the generating series for the set $\mathcal{S}$ with respect to the number of transitions between states. We denote this generating series by $H(y_{0,0}, y_{0,1}, y_{1,0}, y_{1,1})$, which is defined as:

$$H(y_{0,0}, y_{0,1}, y_{1,0}, y_{1,1}) \triangleq [(\mathcal{S}, w_{0,0} \otimes w_{0,1} \otimes w_{1,0} \otimes w_{1,1})] = \sum_{\sigma \in \mathcal{S}} y_{0,0}^{w_{0,0}(\sigma)} y_{0,1}^{w_{0,1}(\sigma)} y_{1,0}^{w_{1,0}(\sigma)} y_{1,1}^{w_{1,1}(\sigma)}; \tag{16}$$

An explicit expression for $H(y_{0,0}, y_{0,1}, y_{1,0}, y_{1,1})$ can be obtained from Equation (15). The adjacency matrix B, therefore, is given by:

$$B = \begin{bmatrix} y_{0,0} & y_{0,1} \\ y_{1,0} & y_{1,1} \end{bmatrix}. \tag{17}$$

Thus

$$\begin{aligned} H(y_{0,0}, y_{0,1}, y_{1,0}, y_{1,1}) &= \operatorname{trace}((I - B)^{-1} J) \\ &= \frac{1 - y_{0,0} + y_{0,1} + y_{1,0} + 1 - y_{1,1}}{1 - y_{1,1} - y_{0,0} + y_{0,0}\, y_{1,1} - y_{1,0}\, y_{0,1}}. \end{aligned} \tag{18}$$

We may also be interested in enumerating a subset of $\mathcal{S}$. Consider the subset $\mathcal{S}_i$, $i = 0, 1$, such that a sequence $\sigma = (\sigma_1, \sigma_2, \ldots)$ is in $\mathcal{S}_i$ if $\sigma_1 = i$,i.e., the first symbol of σ is i.

Defining $H_i(y_{0,0}, y_{0,1}, y_{1,0}, y_{1,1}) = [(\mathcal{S}_i, w_{0,0} \otimes \ldots \otimes w_{1,1})]$, $i = 0, 1$, we obtain:

$$H_0(y_{0,0}, y_{0,1}, y_{1,0}, y_{1,1}) = [(I - B)^{-1}]_{0,0} + [(I - B)^{-1}]_{0,1} = \frac{y_{0,1} + 1 - y_{1,1}}{1 - y_{1,1} - y_{0,0} + y_{0,0}\, y_{1,1} - y_{1,0}\, y_{0,1}} \tag{19}$$

and

$$H_1(y_{0,0}, y_{0,1}, y_{1,0}, y_{1,1}) = [(I - B)^{-1}]_{1,0} + [(I - B)^{-1}]_{1,1} = \frac{1 - y_{0,0} + y_{1,0}}{1 - y_{1,1} - y_{0,0} + y_{0,0}\, y_{1,1} - y_{1,0}\, y_{0,1}} \tag{20}$$

The above equations enumerate the set of all possible transitions of states and a wide class of problems can be solved as a specialization of (19) and (20). In the next section, we will derive a new closed form expression for $p_B(k, n)$, the probability of k transitions from the burst state occurring in a sequence of length n, from the enumeration approach derived in this section.

4 Determination of $p_B(k, n)$

Notice that $p_B(k, n)$ can be expressed as:

$$p_B(k, n) = p\Big(\bigcup_{\sigma \in \mathcal{N}_2^\star} \sigma \text{ such that } w_{0,1}(\sigma) + w_{1,1}(\sigma) = k,\ \sum_{i,j=0}^{1} w_{i,j}(\sigma) = n\Big). \tag{21}$$

The expression above suggests that $p_B(k,n)$ may be formulated as the coefficient of the generating series $H_B(s,z)$, that is:

$$p_B(k,n) = [s^k\, z^n]\, H_B(s,z), \tag{22}$$

where

$$\begin{aligned} H_B(s,z) &= \sum_{\sigma\in\mathcal{N}_2^\star} p(\sigma)\, s^{w_{0,1}(\sigma)+w_{1,1}(\sigma)}\, z^{w_{0,0}(\sigma)+w_{0,1}(\sigma)+w_{1,0}(\sigma)+w_{1,1}(\sigma)} \\ &= \sum_{k,n} p_B(k,n)\, s^k\, z^n \qquad \in\ R[s][[z]], \end{aligned} \tag{23}$$

and the indeterminate z marks the length of σ and s marks the number of transitions from state 1. The generating series $H_B(s,z)$ is a polynomial in z with a coefficient ring $R[s]$. The series $H_B(s,z)$ may be expressed in terms of $H_0(y_{0,0}, y_{0,1}, y_{1,0}, y_{1,1})$ and $H_1(y_{0,0}, y_{0,1}, y_{1,0}, y_{1,1})$ defined in Equations (19) and (20) by applying the mapping:

$$\begin{aligned} y_{0,0} &\mapsto (1-Q)z \\ y_{0,1} &\mapsto Qsz \\ y_{1,0} &\mapsto qz \\ y_{1,1} &\mapsto (1-q)sz. \end{aligned} \tag{24}$$

So

$$\begin{aligned} p_B(k,n) &= p_B(k,n|S_0=0)\, p(S_0=0) \;+\; p_B(k,n|S_0=1)\, p(S_0=1) \\ &= [s^k\, z^n]\; H_0(\,(1-Q)z, Qsz, qz, (1-q)sz)\, p(S_0=0) + \\ &\qquad\qquad H_1(\,(1-Q)z, Qsz, qz, (1-q)sz)\, p(S_0=1). \end{aligned} \tag{25}$$

Upon substitution of (19) and (20) into (25) yields

$$p_B(k,n) = [s^k\, z^n]\, \frac{(q+Q) - q(1-q-Q)sz - Q(1-q-Q)z}{(q+Q)[1-(1-q)sz-(1-Q)z+(1-q-Q)sz^2]}. \tag{26}$$

Notice that $p_B(k,n)$ can be expressed in a matrix form:

$$p_B(k,n) = [s^k\, z^n] \begin{bmatrix} \frac{q}{Q+q} & \frac{Q}{Q+q} \end{bmatrix} \begin{bmatrix} 1-(1-Q)z & Qsz \\ qz & 1-(1-q)sz \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{27}$$

The task that remains is the extraction of the coefficient $[s^k\, z^n]$ in Equation (26). This can be accomplished using the partial fraction technique and leads to a closed form solution for $p_B(k,n)$ (see Appendix A for details). The final expression is

$$p_B(k,n) \;=\; g(k,n) \;-\; \frac{q(1-q-Q)}{q+Q}\, g(k-1,n-1) \;-\; \frac{Q(1-q-Q)}{q+Q}\, g(k,n-1), \tag{28}$$

where

$$g(k,n) \;=\; \frac{(1-q)^k(1-Q)^{n-k}}{2^n} \sum_{\ell\geq 0}\sum_{j\geq 0} \binom{n+1}{2j+1} \binom{j}{l} \binom{n-2l}{k-l} \frac{[-4(1-q-Q)]^\ell}{[(1-q)(1-Q)]^\ell}. \tag{29}$$

The problem of finding an explicit expression for $p_B(k,n)$ was solved by finding the power series expression of the function that appears in Equation (26). We can also obtain an alternative expression for $p_B(k,n)$ in a form of a recurrence equation. From Equation (26), the following recurrence is valid for $p_B(k,n)$

$$p_B(k,n) - (1-q)\, p_B(k-1,n-1) - (1-Q)\, p_B(k,n-1) + (1-q-Q)\, p_B(k-1,n-2) = 0, \tag{30}$$

with the initial conditions $p_B(0,0)=1$, $p_B(0,1)=\frac{q}{q+Q}$, $p_B(1,0)=\frac{Q}{q+Q}$.

5 Number of distinct burst intervals

The Gilbert-Elliott channel can be typically pictured as the channel with two operating modes. There are extended time intervals displaying a relatively low error rate, separated by error bursts or intervals with much higher error rate. The model is able to describe the burst nature of channel because the probabilities of changing states are small. We now address the problem of determining the number of distinct burst intervals for the Gilbert-Elliott channel. This problem arises when we wish to design coding for Gilbert-Elliott channels with multiple burst error correction capability. The number of distinct burst intervals for a given sequence is defined as the number of distinct intervals in the burst state. For example, the state sequence $\sigma = (0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0,1,1)$ has two burst intervals. Let the random variable X_n be the number of burst intervals in a sequence of length n. The random variable X_n is related to the random variable Y_n, the number of transitions from state 0 to 1, by the following relation:

$$p(X_n = k) = p(Y_n = k|S_1 = 0)\,p(S_1 = 0) + p(Y_n = k-1|S_1 = 1)\,p(S_1 = 1). \tag{31}$$

The probability $p(Y_n = k \mid S_1 = i)$, $i = \{0,1\}$ can be calculated using the enumerative technique described in the last section. Let the indeterminate z mark the length of σ and let s mark the number of transitions from state 0 to state 1. Following the same formulation that led to Equation (25), we express $p(X_n = k)$ as the coefficient of a generating series. Then

$$\begin{aligned} p(X_n = k) &= [s^k z^{n-1}]\, H_0(\,(1-Q)z, Qsz, qz, (1-q)z)\,p(S_0 = 0) + \\ &\quad [s^{k-1} z^{n-1}]\, H_1((1-Q)z, Qsz, qz, (1-q)z)\,p(S_0 = 1); \\ &= [s^k z^{n-1}]\, H_0(\,(1-Q)z, Qsz, qz, (1-q)z)\,p(S_0 = 0) + \\ &\quad [s^k z^{n-1}]\, s\, H_1((1-Q)z, Qsz, qz, (1-q)z)\,p(S_0 = 1), \end{aligned} \tag{32}$$

where the functions $H_0(y_{0,0}, y_{0,1}, y_{1,0}, y_{1,1})$ and $H_1(y_{0,0}, y_{0,1}, y_{1,0}, y_{1,1})$ are given by Equations (19) and (20). It can be shown that $p(Y_n = k)$ is given by

$$p(X_n = k) = [s^k z^{n-1}]\, \frac{q - q(1-q)z + Qs - (Q(1-Q) - 2Qq)sz}{(q+Q)[1 - (2-q-Q)z + (1-Q)(1-q)z^2 - Qqsz^2]}. \tag{33}$$

The partial fraction technique described in the Appendix A can be used to extract the coefficient of Equation (33). Alternatively, a recurrent formula can also be easily obtained.

Figure 2 shows $p(X_n = k)$ as a function of k for the Gilbert-Elliott channel with parameters Q=0.02, q=0.1, for n=100 and n=150. For the set of parameters considered, the plots show that sequences with 2 bursts occur with the highest probability for n=100. Sequences with 3 bursts occur more likely for $n = 150$. For the two values of n considered, the probability of more than 10 bursts is less than 10^{-5}.

6 Conclusions

In this paper we have derived the generating series for all possible transition of states of a Markov chain with an arbitrary number of states. We used this enumerative approach to find new closed form solutions and recurrence equations for some statistics of interest in the analysis of Gilbert-Elliott channels. Two statistics were considered; the number of transitions from the bad state, and the number of burst intervals. These statistics can

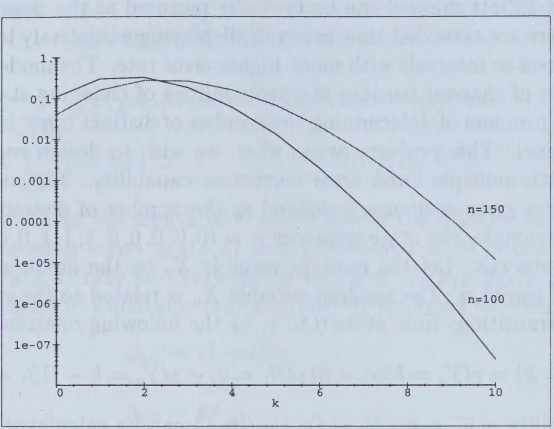


Figure 2: Probability of k distinct burst intervals for sequences of length n, as a function of k, for n=100 and n=150.

be used to design and analyze encoders and interleavers for finite state channels. Similar problems involving Markov chains with larger number of states may also be addressed with the technique presented in this paper.

A Appendix

This appendix contains an outline of the extraction of the coefficient $[s^k\, z^n]$ of Equation (26). For convenience we repeat this equation below.

$$p_B(k,n) = [s^k\, z^n]\, \frac{(q+Q) - q(1-Q-q)sz - Q(1-Q-q)z}{(q+Q)[1-(1-q)sz-(1-Q)z+(1-Q-q)sz^2]}, \tag{34}$$

First let the function $G(s,z)$ be defined as:

$$G(s,z) = \frac{1}{1-(1-q)sz-(1-Q)z+(1-Q-q)sz^2}, \tag{35}$$

So from Equation (34)

$$\begin{aligned} p_B(k,n) &= [s^k\, z^n]\, G(s,z) - \tfrac{q(1-q-Q)}{q+Q}\, [s^{k-1}\, z^{n-1}] G(s,z) - \tfrac{Q(1-q-Q)}{q+Q}\, [s^k\, z^{n-1}] G(s,z) \\ &= g(k,n) - \tfrac{q(1-q-Q)}{q+Q}\, g(k-1,n-1) - \tfrac{Q(1-q-Q)}{q+Q}\, g(k,n-1), \end{aligned} \tag{36}$$

where

$$g(k,n) = [s^k\, z^n]\, G(s,z). \tag{37}$$

The rest of this appendix is dedicated to find an explicit expression for $g(k,n)$. The partial fraction expression of $G(s,z)$ in (35) has the form

$$G(s,z) = \frac{c_1(s)}{1-zf_1(s)} + \frac{c_2(s)}{1-zf_2(s)}, \tag{38}$$

where

$$\begin{aligned} f_1(s) &= \tfrac{\alpha-\gamma}{2} \\ f_2(s) &= \tfrac{\alpha+\gamma}{2} \\ c_1(s) &= -\tfrac{\alpha-\gamma}{2} \\ c_2(s) &= \tfrac{\alpha+\gamma}{2} \\ \alpha &= (1-Q)+(1-q)s \\ \gamma &= \sqrt{\alpha^2-4(1-q-Q)s}. \end{aligned} \tag{39}$$

Applying the coefficient operator $[z^n]$ to Equation (38) gives

$$\begin{aligned} g(k,n) &= [s^k]\,[c_1(s)\,f_1^n(s) \,+\, c_2(s)\,f_2^n(s)] \\ &= [s^k]\,\tfrac{1}{2^{n+1}\gamma}[(\alpha+\gamma)^{n+1}-(\alpha-\gamma)^{n+1}]. \end{aligned} \tag{40}$$

Since

$$(\alpha+\gamma)^m = \sum_{i=0}^{m}\binom{m}{i}\gamma^i\,\alpha^{m-i}. \tag{41}$$

Then

$$\begin{aligned} g(k,n) &= [s^k]\,\tfrac{1}{2^{n+1}\gamma}\sum_{i=0}^{n+1}\tbinom{n+1}{i}\alpha^{n+1-i}\,(\gamma^i-(-\gamma)^i) \\ &= [s^k]\,\tfrac{1}{2^n\gamma}\sum_{j\geq 0}\tbinom{n+1}{2j+1}\alpha^{n-2j}\,\gamma^{2j+1} \\ &= [s^k]\,\tfrac{1}{2^n}\sum_{j\geq 0}\tbinom{n+1}{2j+1}\alpha^{n-2j}\,(\alpha^2-4(1-q-Q)s)^j \\ &= [s^k]\,\tfrac{1}{2^n}\sum_{j\geq 0}\sum_{\ell\geq 0}\tbinom{n+1}{2j+1}\tbinom{j}{\ell}(-4(1-q-Q)s)^\ell\alpha^{n-2\ell}. \end{aligned} \tag{42}$$

Using the identity

$$\alpha^{n-2\ell} = [(1-Q)+(1-q)s]^{n-2\ell} = \sum_{r\geq 0}\binom{n-2\ell}{r}[(1-q)s]^r(1-Q)^{n-2\ell-r}. \tag{43}$$

So

$$g(k,n) = [s^k]\frac{1}{2^n}\sum_{j\geq 0}\sum_{\ell\geq 0}\sum_{r\geq 0}\binom{n+1}{2j+1}\binom{j}{\ell}\binom{n-2\ell}{r}(-4(1-q-Q))^\ell(1-q)^r\,(1-Q)^{n-2\ell-r}\,s^{\ell+r}. \tag{44}$$

The coefficient $[s^k]$ can be determined by carrying out the summation on ℓ and r such that $\ell+r=k$. Thus Equation (29) follows.

References

[1] L. N. Kanal and A. Sastry, “Models for channels with memory and their applications to error control,” *Proceedings of the IEEE*, vol. 66, pp. 724–744, July 1987.

[2] E. O. Elliott, “Estimates of error rates for codes on burst-noise channels,” *The Bell System Technical Journal*, vol. 42, pp. 1977–1997, September 1963.

[3] J. R. Yee and E. J. W. Jr., “Evaluation of the performance of error-correcting codes on a gilbert channel,” *IEEE Transactions on Communications*, vol. 43, pp. 2316–2323, August 1995.

[4] I. P. Goulden and D. M. Jackson, *Combinatorial Enumeration.* John Wiley & Sons, 1983.

Nonrigid Motion Tracking

Chandra Kambhamettu
Universities Space Research Association
Mesoscale Dynamics and Precipitation Branch
NASA/Goddard Space Flight Center
Greenbelt, MD 20771, USA
(tel) 301 286-5786, (e-mail) chandra@gloria.gsfc.nasa.gov

Abstract

Recovery of motion parameters and point correspondences is a fundamental problem in computer vision. Nonrigid motion analysis, in particular, is gaining interest due to its wide range of applications. In this paper, we present a novel method for estimating motion parameters and point correspondences between 3D surfaces undergoing nonrigid motion. We present a multi-resolution, differential geometry-based system, where different nonrigid motion models are arranged in a coarse-to-fine fashion. The motion parameters and correspondences will be refined at every step in the hierarchy, as each motion model searches for a better match with the previous estimation as its input. Invariant features of surfaces such as Gaussian curvature, unit-normal and discriminant are used in the analysis. The motion parameter used in our approach is a vector point function, called *displacement function.* It represents both direction and magnitude of deformation at a given point. Potential applications for this system are numerous, including the fields of bio-medical image analysis, high-speed teleconferencing and meteorology. In this paper, we present experiments on facial and cloud motion tracking, demonstrating performance of the system.

1 Introduction

Motion perception in humans is a natural, well-developed visual capability that is an essential survival skill. Humans are able to categorize the relative motions between different objects in a scene with respect to an arbitrary co-ordinate system quite rapidly and intuitively. Yet, developing automatic computational motion analysis algorithms capable of handling nonrigid motion estimation has been a difficult task. One common problem of any motion analysis algorithm is to determine correspondences between various regions of the image in a time sequence. Ideally, estimating point correspondence at every point of the image for a dense flow field will be very useful. There are numerous areas where nonrigid motion estimation can be applied including the atmospheric sciences, medicine, robotics and manufacturing. Over the last two decades there have been numerous developments in the area of motion analysis [7, 27].

Most of the work in motion analysis is based on the rigidity assumption where the shape of objects do not change over time. The rigidity assumption fails in numerous motion analysis situations, as many real-world objects are nonrigid. Typical examples of nonrigid motion behavior are prevalent in nature including the dynamics of clouds and aerosols, dynamics of water waves and currents, dynamics of sand and soil, motion of animals, plants and biological cells, or the deformation of flexible structures and industrial components. Fluid motion is perhaps the most complex that is commonly observed. There is usually no continuity constraint among neighboring particles since they move freely according to the underlying dynamics. Cloud motion is a special case of nonrigid motion, where there is partial fluid and partial solid motion, called semi-fluid motion [20]. There is a growing need for motion analysis algorithms, designed to tackle the more complex categories of nonrigid motion.

Motion by itself has different variations of complexity. Rigid motion can be considered as the simplest of all, and fluid motion the most complex of all. It is important to categorize motion depending on the problem at hand, so as to choose the appropriate method of analysis [8, 13].

Rigid motion preserves the 3-D distances between any two points of an object, where the transformations only include rotation and translation. The object does not stretch or bend, hence some of the differential geometric invariants remain constant. For example, curvature, fundamental forms, etc. can be used to track the motion. A great deal of research has been done on this category of motion with the additional emphasis on also extracting structure from motion [21, 7, 23]. *Articulated motion* is piecewise rigid motion. This class of nonrigid motion of an object is closest to rigid motion behavior, where the object's rigid parts are connected by nonrigid joints. Examples of this type of motion include animal (skeletal) body motion and robot manipulators [30, 6, 19, 22]. *Quasi-rigid motion* restricts the deformation to be small. A general nonrigid motion is quasi-rigid when viewed in a sufficiently short time interval, say between image frames when the sampling rate is high enough [29, 25, 9]. *Isometric motion* is nonrigid motion which preserves lengths along the surface as well as angles between curves on the surface [15, 10, 3]. *Homothetic motion* is a uniform expansion or contraction of a surface [5, 11]. *Conformal motion* is nonrigid motion which preserves angles between curves on the surface during motion, but not lengths [17, 18, 1]. *Elastic motion* is a nonrigid motion whose only constraint is some degree of continuity or smoothness. Elastic motion of solid objects has been found to have a wide variety of real-world applications and as such has been the focus of a great deal of research [28, 14, 26, 2, 16, 24, 31, 8, 12]. *Semi-fluid motion* involves partial fluidity, where the continuity constraint can be relaxed locally [20]. *Fluid motion* is a general nonrigid motion that need not be continuous. This is the most difficult category of nonrigid motion to analyze since it may involve topological variation and turbulent deformation behaviors [8, 4, 32].

Among the motion categories described above, elastic motion, semi-fluid motion and fluid motion are the more general categories of nonrigid motion. There has been an increasing degree of interest and research in the individual categories of motion behavior, however, there have not been many attempts to combine different classes of motion models into one integrated system. In this work, we combine various classes of motion into a single hierarchical structure, arranged in a *coarse-to-fine* fashion. This will allow for a multi-resolution motion matching, where the analysis is refined iteratively at each level.

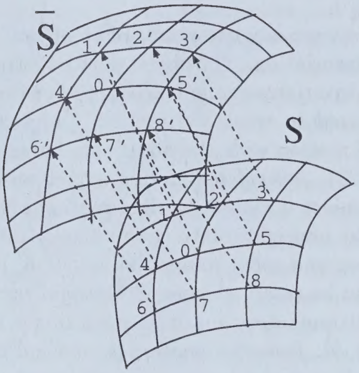

$\bar{S}$
1′
2′
3′
4
0′
5′
8′
S
6′
7′
3
2
1
5
0
4
8
6
7

Figure 1: Correspondences in 3D surfaces

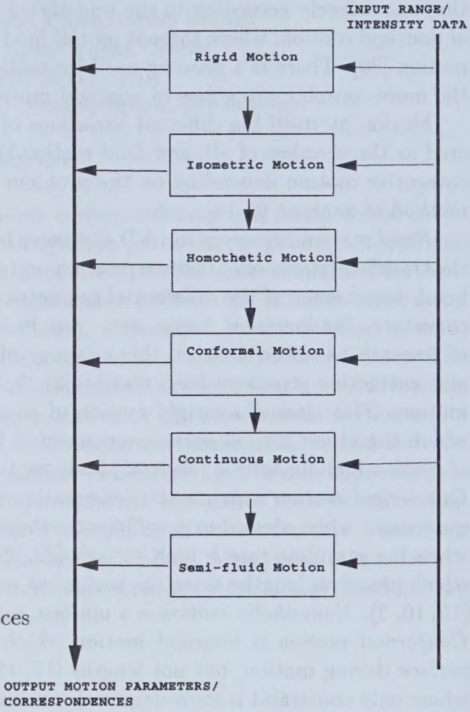

INPUT RANGE/
INTENSITY DATA
Rigid Motion
Isometric Motion
Homothetic Motion
Conformal Motion
Continuous Motion
Semi-fluid Motion
OUTPUT MOTION PARAMETERS/
CORRESPONDENCES

Figure 2: Multi-resolution Motion Analysis System

2 Multi-resolution Motion analysis

In this section, we present an integrated system containing different motion models which have been presented in the previous section. Our problem is to estimate motion parameters and point correspondences between a surface and its deformed counterpart as shown in Figure 1. The possible correspondences for certain points on the surface were shown by the arrows in the figure.

Figure 2 represents an integrated system, having different motion models arranged in a coarse-to-fine fashion. Each module consists of an error measure based on differential geometric relation. We minimize this error to estimate for the motion parameter and point correspondence. The motion parameter used is a displacement function, represented by $s(x, y)$. Any point on the surface S, given by $P(x, y, z)$, can be transformed to $P'(x', y', z')$ due to nonrigid motion. The displacement function is then given by,

$$\mathrm{s} = (x' - x, y' - y, z'(x', y') - z(x, y)). \tag{1}$$

We now present error measures that were used to estimate motion parameters and point correspondences at each module. The error measure is represented by ER, which is expressed in terms of differential geometric parameters such as Gaussian curvature (K), Unit-normal (n), coefficients of first fundamental form (E, F, G) and Discriminant (D). Note that the parameters corresponding to the deformed surface are represented by primes such as K' for Gaussian curvature, n' for unit-normal etc. Each module is based on geometry only, and the total number of modules can be extended to include other motion models. For example, a physics-based motion analysis module can be appended at the end of the hierarchy.
Rigid Motion: $ER = (K' - K)^2 + (H' - H)^2$
Isometric Motion: $ER = (K' - K)^2$
Homothetic Motion (t is stretching at each point): $ER = (K't^2 - K)^2$
Conformal Motion: $ER = (E' - t^2E)^2 + (F' - t^2F)^2 + (G' - t^2G)^2$
Continuous Motion (s is displacement function at each point): $ER = (n' - n \times \mathbf{rot}s)^2$
Semi-fluid Motion (This is an extension of the continuous motion, where *semi-fluidity* is defined using the following discriminant measure. Please refer to [20] for more description of this model): $ER_D = (D' - D\, \mathbf{Div}s)^2$

Input to this system, shown in Figure 2 can be a pair of range, intensity, or both sets of images. Images are input into the system at any level, and the output can be extracted at any level. In order to estimate point correspondence, error measure (ER) is applied for each correspondence hypothesis over a template map. Each hypothesis correspondence will have a corresponding error measure, the minimum of which represents a good estimate for the correspondence. More than one correspondence can also be estimated with confidence measures for each. ER represents confidence measure when the underlying motion assumption is followed. Depending on this value, the estimation can be further refined, or stopped at any given level. In the next sections, we present our experimental results on facial and cloud motion.

3 Experiments

In this section, experiments are presented on facial data, and cloud data sets.

3.1 Facial Motion

Our emphasis in this section is on tracking elastic (continuous) surfaces. Input to our algorithm in this experiment is two frames of range data of face (data is provided to us by Dr. Thomas Huang of Univ. of Illinois at Urbana). First frame (Figure 3) is a normal face (without any specific expression), and the second frame (Figure 4) has smile expression of the same subject. Figure 5 depicts some of the point correspondences at important features. Point correspondences on only one side of the face are depicted so as to improve visualization. It can be observed that the correspondences conform with the movement of facial skin during smile. For example, the lips are opened up (and stretched), and also there is a bulge on the cheeks. There is no motion on rigid parts such as nose and corner of eye. Some of the feature points whose correspondences

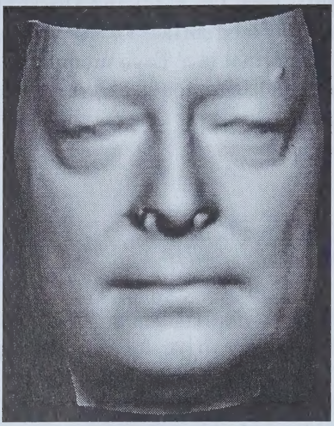


Figure 3: Normal Face

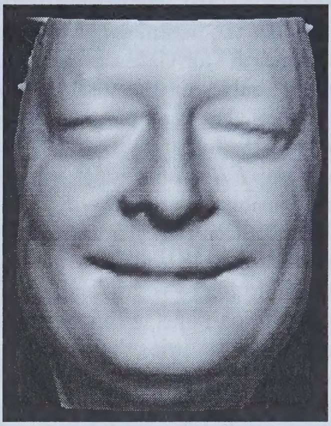


Figure 4: Smiling Face

are shown here are, tip of the nose (rigid), bulging cheek (nonrigid), opening of lips (nonrigid), eye corner (rigid) and chin (nonrigid). Next, we interpolate the intermediate frames using estimated point correspondence information. The interpolated frames are shown in Figures 6 to 13, where the intermediate frames are generated by simply plotting data at various intervals along the point correspondence vectors. There are numerous applications for such intermediate frame-data generation. One immediate use is the smooth animation of nonrigid objects. Our contribution stands out in that, we preserve the semantic annotation while interpolating. For example, observe the smooth transition from *no-smile* to *smile* expression in the interpolated frames (Figures 6 to 13).

3.2 Cloud Motion

We now demonstrate the working of our integrated multi-resolution system on Hurricane Frederic dataset. Stereo image pairs over several time steps, acquired by two geostationary satellites with synchronized scanning instruments are used in our analysis to generate cloud-top height estimation and cloud wind measurements. Four time sequential stereo pairs with a time interval of 7.5 minutes between each stereo pair and a lag of 15 seconds between the left and right pair were used. We have first generated the four disparity maps corresponding to each of the four time frames. Motion analysis was initiated with rigid matching, and then gradually refined to semi-fluid motion analysis. The final results compare well with the manual analysis of cloud speed, performed by an expert in the field. Less than 10% error was observed in wind speed computations, which can be attributed to human error in manual tracking. Figure 14 is the intensity image of hurricane Frederic (Frame-3), with dots representing tracked points. Figure 15 represents the intensity image at Frame-4 with intermediate results of tracked points. One can observe that the point correspondence estimation failed at the cloud tracer pointed by the arrow. This error was corrected by passing the intermediate result as

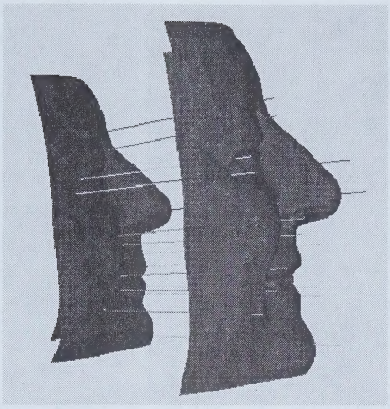


Figure 5: Feature Correspondences

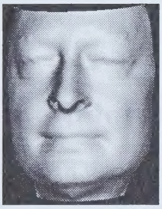


Figure 6: Face-1

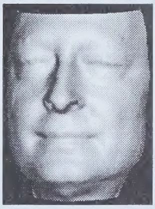


Figure 7: Face-2

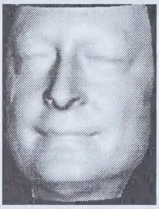


Figure 8: Face-3

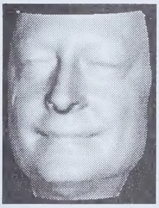


Figure 9: Face-4

input to a motion model at a finer resolution (in this case, semi-fluid motion analysis). The output of this level is shown in Figure 16, where we can clearly observe better estimation of point correspondences.

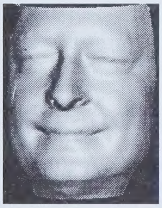


Figure 10: Face-5

Figure 11: Face-6

Figure 12: Face-7

Figure 13: Face-8

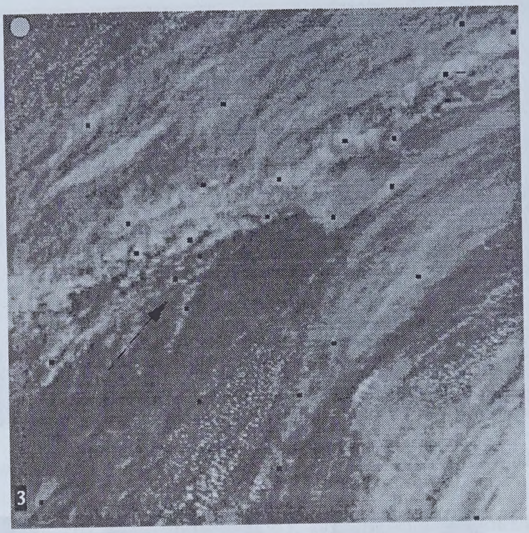


Figure 14: Hurricane Frederic image at t3

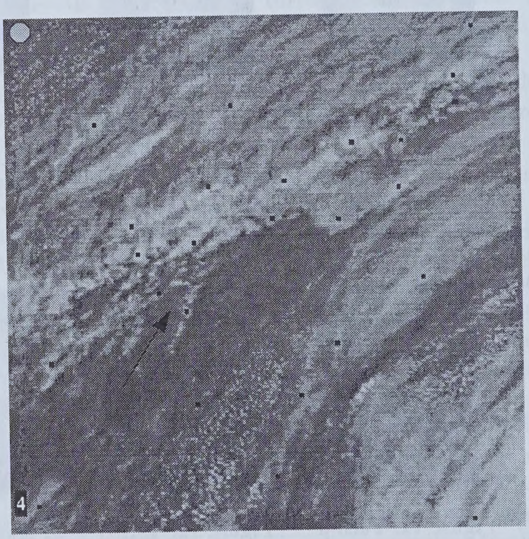


Figure 15: Intermediate result

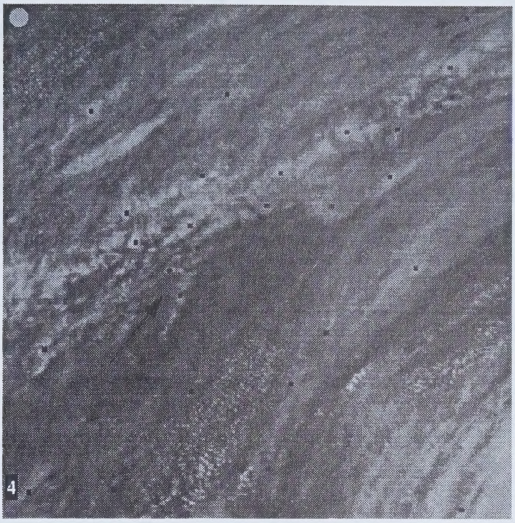


Figure 16: Final result

4 Conclusions

An integrated system for multi-resolution motion analysis is presented based on differential geometry. The system is capable of handling varied categories of motion, arranged in a multi-resolution hierarchy. We have demonstrated the quality performance of the system on facial motion, and complex structures such as clouds. Results of the multi-resolution motion analysis algorithm is also very promising and is comparable to the human analysis of cloud motion. New directions for incorporating robustness, motion segmentation, adaptive searching is necessary for enhancing the integrated system. Our future work includes a preprocessing phase for motion segmentation using expert knowledge. Extensions to the system include incorporating various other motion models at different resolution levels. Model-based and physics-based analysis of motion is another direction to improve on the existing integrated system.

References

[1] Amir A. Amini and James S. Duncan. Pointwise tracking of left-ventricular motion in 3D. *Proceedings of the IEEE Workshop on Visual Motion*, pages 294–299, 1991.

[2] H. I. Bozma and J. S. Duncan. Model-based recognition of multiple deformable objects using a game-theoretic framework. *12'th International Conference, IPMI'91*, pages 358–372, July 7-12 1991.

[3] S. S. Chen and M. Penna. Motion analysis of deformable objects. In T. S. Huang, editor, *Advances in Computer Vision and Image Processing*, volume 3. JAI Press, 1988.

[4] M.S. Chong, Perry A. E., and Cantwell B. J. A general classification of three-dimensional flow fields. *Phys. Fluids A., Vol. 2*, pages 765–777, May, 1990.

[5] D. B. Goldgof, H. Lee, and T. S. Huang. Motion analysis of nonrigid surfaces. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 375–380, 1988.

[6] D. D. Hoffman and B. E. Flinchbaugh. The interpretation of biological motion. *Biol. Cybernet.*, 42:195–204, 1982.

[7] T. S. Huang. Motion analysis. In *Encyclopedia of Artificial Intelligence*, volume 1, pages 620–632. John Wiley and Sons, New York, 1986.

[8] T. S. Huang. Modeling,analysis,and visualization of nonrigid object motion. *Proceedings of 10th ICPR*, pages 361–364, dec 1990.

[9] Ramesh Jain and I. K. Sethi. Establishing correspondence of non-rigid objects using smoothness of motion. *Workshop on computer vision: Representation and Control*, pages 83–87, 1984.

[10] R. Jasinschi and A. Yuille. Nonrigid motion and regge calculus. *J. Opt. Soc. Am.*, A6:1088–1095, 1989.

[11] Chandra Kambhamettu and Dmitry B. Goldgof. Point correspondence recovery in nonrigid motion. *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 222–227, June 1992.

[12] Chandra Kambhamettu, Dmitry B. Goldgof, and Matthew He. Determination of motion parameters and estimation of point correspondences in small nonrigid deformations. *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 943–946, June 1994.

[13] Chandra Kambhamettu, Dmitry B. Goldgof, Demetri Terzopoulos, and Thomas S. Huang. Nonrigid motion analysis. In Tzay Young, editor, *Handbook of PRIP: Computer vision*, volume II, pages 405–430. Academic Press, San Diego, California, 1994.

[14] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[15] J. J. Koenderink. Shape from motion and bending transformation. *Jornal of Optical Society of America*, A1:1265–1266, 1984.

[16] D. Metaxas and D. Terzopoulos. Shape and nonrigid motion estimation through physics-based synthesis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15, 1993. Also in Proc. IEEE Computer Vision and Pattern Recognition Conference (CVPR'91), pp. 337–343, Hawaii, June 1991.

[17] Sanjoy K. Mishra, Dmitry B. Goldgof, and Thomas S. Huang. Non-rigid motion analysis and epicardial deformation estimation from angiography data. *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 331–336, June 1991.

[18] Sanjoy K. Mishra, Dmitry B. Goldgof, and Chandra Kambhamettu. Estimating non-rigid motion from point and line correspondences. *Pattern Recognition Letters*, 15(6):559–566.

[19] J. O'Rourke and N. Badler. Model-based image analysis of human motion using constraint propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2:522–536, 1980.

[20] K. Palaniappan, Chandra Kambhamettu, A. Frederick Hasler, and Dmitry B. Goldgof. Structure and semi-fluid motion analysis of stereoscopic satellite images for cloud tracking. *International Conference on Computer Vision*, pages 659–665, 1995.

[21] J. W. Roach and J. K. Aggarwal. Determining the movement of objects from a sequence of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(6):554–562, 1980.

[22] Maha Sallam and Kevin Bowyer. Generalizing the aspect graph concept to include articulated assemblies. *Pattern Recognition Letters*, 12:171–176, March 1991.

[23] Guna S. Seetharaman. Nonrigid motion analysis. In Tzay Young, editor, *Handbook of PRIP: Computer vision*, volume II, pages 361–403. Academic Press, San Diego, California, 1994.

[24] Jakub Segen and Kristin Dana. Parallel symbolic recognition of deformable shapes. In H. Burkhardt, Y. Neuvo, and J.C. Simon, editors, *From Pixels to Features II*, pages 387–400. Elsevier Science Publishers B.V. (North-Holland), 1991.

[25] D. Shulman and J.Y. Aloimonos. (non-)rigid motion interpretation: a regularized approach. *Proceedings of Royal Society of London*, B233:217–234, 1988.

[26] R. Szeliski and D. Terzopoulos. Physically-based and probabilistic modeling for computer vision. In B. C. Vemuri, editor, *Proc. SPIE 1570, Geometric Methods in Computer Vision*, pages 140–152. Society of Photo-Optical Instrumentation Engineers, San Diego, CA, July 1991.

[27] Richard Szeliski and James Coughlan. Hierarchical spline-based image registration. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 194–201, June 1994.

[28] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: Recovering 3d shape and nonrigid motion. *Artificial Intelligence*, 36(1):91–123, 1988.

[29] S. Ullman. Maximizing rigidity: the incremental recovery of 3-D structure from rigid and nonrigid motion. *Perception*, 13:255–274, 1984.

[30] J. A. Webb and J. K. Aggarwal. Structure from motion of rigid and jointed objects. *Artificial Intelligence*, 19(1):107–130, 1983.

[31] Masanobu Yamamoto, Pierre Boulanger, J. Angelo Beraldin, and Marc Rioux. Direct estimation of range flow on deformable shape from a video rate range camera. *Transactions on Pattern Analysis and Machine Intelligence*, 15(1):82–89, 1993.

[32] Jialin Zhong, Thomas Huang, and Ronald Adrian. Salient structure analysis of fluid flow. *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 310–315, June 1994.

A Systematic approach to reason about actions, planning and robot control

Chitta Baral
Department of Computer Science
University of Texas at El Paso
El Paso, TX 79968
chitta@cs.utep.edu

Abstract

We extend the systematic approach to reason about actions to include robot control. We review the high level specification languages $\mathcal{A}$ [GL93] and $\mathcal{L}_0$ [BGP95] and extend them so as to be able to specify robot actions and robot environments through expressions about values of fluents during the execution of an action, failure of the execution of an action, verification and sensing actions, etc.

1 Introduction

Research in theories of actions is concerned with developing formal theories that allows us to represent effects of actions on the world and reason about them. This led to the discovery and consideration of several aspects such as inertia (and the associated frame problem [GL93, Bro87]), hypothetical facts [GL93, BGP95] constraints and indirect effects (and the associated qualification and ramification problem [LR94, KL94, Bar95]), non-deterministic and other complex effects of actions [Bar95, Pin94], narratives and actual situations [MS94, PR93, BGP95], concurrent and compound actions[LLL$^+$94, LS92, BG93], causality and dependency between fluents[MT95, Lin95, Bar95, GL95], knowledge producing or sensing actions [SL93], etc.

Until recently research on the above was based on formalizing particular examples in a *logical language* (preferably non-monotonic) such as first-order logic augmented with circumscription, default logic and logic programs, and usually together with situation calculus notation. Recently the field has attained some maturity and the focus has been to generalize this approach either through a direct specification language (such as ADL [Ped89], $\mathcal{A}$ [GL93], and the language in [San92, San94]) or by putting restrictions on the logical language that describes the domain (there by restricting the domain) [Rei91, MS94] (See Kartha [Kar93] for results linking the two approaches.) In the first case a translation to a logical language (usually logic programs [GL93, BG93, Bar95] or situation calculus augmented with circumscription [KL94]) is usually provided and soundness and/or completeness results are given. We refer to the recent approach to develop general theories of reasoning about action as *a systematic approach.*

Although one of the agenda behind research in reasoning about actions is to develop autonomous agents (robots) that among other things (such as sense, reason, act etc.)

can make a plan to achieve certain goals, research in planning has usually concentrated on simple theories of action – mostly STRIPS [FN71] and in some recent cases ADL [Ped89]. In other words recent advances in theories of actions that allows more realistic domains have not impacted research in planning.

The other important aspect in building autonomous agents is to consider the dynamic and imprecise nature of the real world. This has been pointed out by many researchers, particularly the ones that advocate 'situated activity' and emphasize ongoing physical interaction with the environment as the main aspect in designing autonomous agents (See [Mae91] for a collection of papers related to this.). Some advocates of this alternative approach go to the extreme and suggest that reasoning and high level planning may not be at all necessary while some others advocate to combine high level planning and reasoning with situated activity.

In this paper we show how the systematic approach to reasoning about actions can be extended to into account the 'realities' (i.e. the dynamic and imprecise nature) of the world. We take into account aspects such as:
(a) that execution of an action may fail,
(b) the agent may not perform exactly as it is asked to,
(c) the world may change from external influences,
(d) the agent may loose track of its surroundings, etc.

We also briefly discuss how planners corresponding to extended and more realistic theories of actions can be developed.

2 Systematic approach to reason about actions and planning

Before we discuss extending systematic approach to reason about actions to robot control, we will briefly describe the action description languages $\mathcal{A}$ [GL93] and $\mathcal{L}_0$ [BGP95] from the literature. We will then extend $\mathcal{L}_0$ to incorporate aspects related to robot control in a real environment.

2.1 The language $\mathcal{A}$

A description of an action domain in the language $\mathcal{A}$ consists of "propositions" of two kinds. A "v-proposition" specifies the value of a fluent in the initial situation, or after performing a sequence of actions. An "effect proposition" describes the effect of an action on a fluent.

The alphabet of $\mathcal{A}$ consists of two disjoint nonempty sets of symbols $\mathcal{F}$, and $\mathcal{A}$, called *fluent names* and *action names* respectively. A *fluent literal* is either a fluent name or a fluent name preceded by $\neg$. For a fluent literal l, by $\bar{l}$ we denote the fluent literal $\neg l$, and we shorten $\neg\neg l$ to l.

A *v-proposition* is an expression of the form

$$f \textbf{ after } a_1, \ldots, a_m, \tag{1}$$

where f is a fluent literal, and $a_1,\ldots,a_m$ $(m \geq 0)$ are action names.
If $m = 0$, we will write (1) as **initially** f.

An *effect proposition* is an expression of the form

$$a \textbf{ causes } f \textbf{ if } p_1, \ldots, p_n, \tag{2}$$

where a is an action name, and each of $f, p_1, \ldots, p_n$ ($n \geq 0$) is a fluent literal. About this proposition we say that it *describes the effect of a on f*, and that $p_1, \ldots, p_n$ are its *preconditions.* If $n = 0$, we will drop **if** and write simply a **causes** f.

A *domain description*, or simply *domain*, is a set of propositions which does not contain contradictory effect propositions.

A *state* is a set of fluent names. Given a fluent name f and a state σ, we say that f *holds* in σ if $f \in \sigma$; $\neg f$ *holds* in σ if $f \notin \sigma$. A *transition function* is a mapping Φ of the set of pairs (a, σ), where a is an action name and σ is a state, into the set of states. A *structure* is a pair (σ_0, Φ), where σ_0 is a state (the *initial state* of the structure), and Φ is a transition function.

We say that a v-proposition (1) is *true* (*false*) in a structure M if f holds in the state $\Phi(a_k, \Phi(a_{k-1}, \ldots, \Phi(a_1, \sigma_0) \ldots))$ (denoted by $M^{(a_1, \ldots, a_m)}$). In particular, the proposition **initially** f is true in M iff f holds in the initial state of M.

A structure (σ_0, Φ) is a *model* of a domain description D if the following two conditions are satisfied: (i) every v-proposition from D is true in (σ_0, Φ); (ii) for every action a and every state σ, $\Phi(a, \sigma) = \sigma \cup \sigma' \setminus \sigma''$, where σ' (σ'') is the set of fluent names f such that D includes an ef-proposition describing the effect of a on f (respectively, $\neg f$) whose preconditions hold in σ.

The condition (ii) characterizes Φ uniquely. Consequently, different models of the same domain description can differ only by their initial states. A domain description is *consistent* if it has a model, and *complete* if it has exactly one model. A v-proposition is *entailed* by a domain description D if it is true in every model of D.

The language $\mathcal{A}$ can be used several simple examples in reasoning about actions. For example, the Yale Shooting problem can be expresses by following the domain description D_0:

initially *alive*
initially $\neg$*loaded*
load **causes** *loaded*
shoot **causes** $\neg$*alive* **if** *loaded*

It is easy to see that $D_0 \models \neg alive$ **after** *load*, *shoot*.

The language $\mathcal{A}$ is a starting point in the systematic approach to reason about actions. It does not allow actual occurances of actions and observations of fluents. To reason about the real world where actions beyond the control of the robot may happen and where the robot not only plans but also executes them we need to extend $\mathcal{A}$. The following language $\mathcal{L}_0$ [BGP95] was the first extension in this direction.

2.2 The language $\mathcal{L}_0$

The alphabet of $\mathcal{L}_0$ consists of three disjoint nonempty sets of symbols $\mathcal{F}$, $\mathcal{A}$ and $\mathcal{S}$, called *fluents*, *actions*, and *actual situations*. Elements of $\mathcal{A}$ and $\mathcal{S}$ will be denoted by (possibly indexed) letters a and s respectively. We will also assume that $\mathcal{S}$ contains two special situations s_0 and s_N called *initial* and *current* situations.

There are two kinds of propositions in $\mathcal{L}_0$ called causal laws and facts.

- A *causal law* in $\mathcal{L}_0$ is same as an effect proposition in $\mathcal{A}$.
- An atomic *fluent fact* is an expression of the form

$$f \ \textbf{at} \ s \tag{3}$$

where f is a fluent literal and s is a situation. (Unless otherwise stated by situations we will mean actual situations.) The intuitive reading of (3) is "f is observed to be true in situation s".

- An atomic *occurrence fact* is an expression of the form

$$\alpha \ \textbf{occurs\_at} \ s \tag{4}$$

where α is a sequence of actions, and s is a situation. It states that "the sequence α of actions was observed to have occurred in situation s". (We assume that actions in the sequence follow each other immediately).

- An atomic *precedence fact* is an expression of the form

$$s_1 \ \textbf{precedes} \ s_2 \tag{5}$$

where s_1 and s_2 are situations. It states that situation s_2 occurred after situation s_1.

Propositions of the type (1) express general knowledge about effects of actions and hence are referred to as *laws.* Propositions (3), (4) and (5) are called *atomic facts* or *observations.* A *fact* is a propositional combination of atomic facts. A collection of laws and facts is called a *domain description* of $\mathcal{L}_0$. The sets of laws and facts of a domain description D will be denoted by D_l and D_f respectively. We will only consider domain descriptions whose propositions do not contain the situation constant s_N.

To see how the domain descriptions of $\mathcal{L}_0$ can be used to represent knowledge about actions let us consider the following example:

Example 2.1 Suppose that we are given a series of observations about "*Fred*":

(a) when the water pistol was squirted *Fred* was seen to be *alive* and *dry*,

(b) in a later moment a shot was fired at *Fred.*

Suppose also that it is generally known that

(c) *squirting* makes *Fred wet*, (d) *shooting* makes *Fred dead*

The above information can be represented by a domain description D_1 consisting of the following propositions:

$$D_1 = \begin{cases} Facts: & \\ (p1) & alive \ \textbf{at} \ s_0 \\ (p2) & dry \ \textbf{at} \ s_0 \\ (p3) & squirt \ \textbf{occurs\_at} \ s_0 \\ (p4) & s_0 \ \textbf{precedes} \ s_1 \\ (p5) & shoot \ \textbf{occurs\_at} \ s_1 \\ Laws: & \\ (p6) & squirt \ \textbf{causes} \ \neg dry \\ (p7) & shoot \ \textbf{causes} \ \neg alive \end{cases}$$

□

Domain descriptions in $\mathcal{L}_0$ are used in conjunction with the following informal assumptions which clarify the description's meaning:

(a) Changes in the values of fluents can only be caused by execution of actions.

(b) There are no actions except those from the language of the domain description.

(c) There are no effects of actions except those specified by the causal laws of the domain.

(d) No actions occur except those needed to explain the facts in the domain description.

(e) Actions do not overlap or happen simultaneously.

These assumptions give a reasonably good intuitive understanding of domain descriptions of $\mathcal{L}_0$. Consider for instance domain description D_1 from Example 2.1. It is easy to see that D_1 together with assumption (d) implies that *squirt* is the only action which occur between s_0 and s_1 and that *shoot* is the only action which occur between s_1 and s_N. Using D_1 with the assumptions (a) – (e) we can conclude that at moment s_1 *Fred* is *wet* but *alive* while at moment s_N (i.e. at the end of the story) he is *wet* and *dead.*

The semantics of domain descriptions of $\mathcal{L}_0$ precisely specify the sets of acceptable conclusions which can be reached from such descriptions and assumptions (a)-(e).

We assume that states of the world are determined by sets of fluents and that actions executed in a particular state can add and remove such fluents according to the causal laws. The set of all possible behaviors a of dynamic system, satisfying causal laws of a given domain description D, can then be described by a transition diagram with states labeled by the sets of fluents and transitions labeled by actions. To interpret facts from D we need to select the initial situation together with a path in the diagram describing the actual behavior of the system.

As in $\mathcal{A}$, a *state* is a set of fluent names. A *causal interpretation* is a partial function Ψ from sequences of actions to states such that:

(1) Empty sequence [ ] belongs to the domain of Ψ and (2) Ψ is prefix-closed [1].

$\Psi([\ ])$ is called the initial state of Ψ. The partial function Ψ serves as an interpretation of causal laws of D.

Given a fluent f and a state σ, we say that f *holds* in σ (f is *true* in σ) if $f \in \sigma$; $\neg f$ *holds* in σ (f is *false* in σ) if $f \notin \sigma$. The *truth* of a propositional formula with respect to σ is defined as usual.

Now we define effects of actions determined by causal laws of a domain description D and our informal assumptions.

A fluent literal f is an (immediate) **effect** of (executing) a in σ if there is an effect law "a **causes** f **if** $p_1, \ldots, p_n$" in D whose preconditions $p_1, \ldots, p_n$ hold in σ. Let

$E_a^+(\sigma) = \{f : f \in \mathcal{F}$ and f is an effect of a in $\sigma\}$,

$E_a^-(\sigma) = \{f : f \in \mathcal{F}$ and $\neg f$ is an effect of a in $\sigma\}$ and

$Res(a, \sigma) = \sigma \cup E_a^+(\sigma) \setminus E_a^-(\sigma)$.

The following definition captures the meaning of causal laws of D.

---

[1] By "prefix closed" we mean that for any sequence of actions α and action a, if $\alpha \circ a$ is in the domain of Ψ then so is α. (Recall that $\circ$ denotes concatenation, and $\alpha \circ a$ means the sequence of actions where a follows α).

Definition 2.1 A causal interpretation Ψ **satisfies** effect laws of D if for any sequence $\alpha \circ a$ from the language of D $\Psi(\alpha \circ a) = Res(a, \Psi(\alpha))$ if $E_a^+(\Psi(\alpha)) \cap E_a^-(\Psi(\alpha)) = \emptyset$ and undefined otherwise. We say Ψ is a **causal model** of D if it satisfies causal laws of D $\square$

It is easy to see that causal models of domain descriptions are uniquely determined by their initial values: i.e. for any causal models Ψ_1 and Ψ_2 of a domain description D, if $\Psi_1([\,]) = \Psi_2([\,])$ then $\Psi_1 = \Psi_2$.

To continue our definition of the semantics of $\mathcal{L}_0$ consider an arbitrary domain description D and let a causal interpretation Ψ be a causal model of D. To interpret the observations of D we first need to define the meaning of situation constants $s_0, s_1, s_2, \ldots$ from $\mathcal{S}$. To do that we consider a mapping Σ from $\mathcal{S}$ to sequences of actions from the language of D. This mapping will be called *a situation assignment of* $\mathcal{D}$ if it satisfies the following properties:

1. $\Sigma(s_0) = [\,]$ 2. for every $s_i \in \mathcal{S}$, $\Sigma(s_i)$ is a prefix of $\Sigma(s_N)$.

Definition 2.2 An *interpretation* M of $\mathcal{L}_0$ is a pair (Ψ, Σ), where Ψ is a causal model of D, Σ is a situation assignment of $\mathcal{S}$ and $\Sigma(s_N)$ belongs to the domain of Ψ. $\square$

$\Sigma(s_N)$ will be called the *actual path* of M and will often be denoted by Σ_N.

Now we can define truth of facts of D w.r.t. an interpretation M. Facts which are not true in M will be called *false* in M.

Definition 2.3 For any interpretation $M = (\Psi, \Sigma)$

(1) (f **at** s) is *true* in M (or satisfied by M) if f is true in $\Psi(\Sigma(s))$.

(2) (α **occurs_at** s) is true in M if the sequence $\Sigma(s) \circ \alpha$ is a prefix of the actual path Σ_N of M.

(3) (s_1 **precedes** s_2) is true in M if $\Sigma(s_1)$ is a proper prefix of $\Sigma(s_2)$

(4) Truth of non-atomic facts in M is defined as usual. $\square$

A set of facts is true in interpretation M if all its members are true in M.

To complete the definition of model we need only to formalize the following underlying assumption of domain descriptions of $\mathcal{L}_0$:

(d) No actions occur except those needed to explain the facts in the domain description.

This is done by imposing a minimality condition on the situation assignments of $\mathcal{S}$ which leads to the following

Definition 2.4 An interpretation $M = (\Psi, \Sigma)$ will be called a *model* of a domain description D in $\mathcal{L}_0$ if the following conditions are satisfied:

1. Ψ is a causal model of D. 2. Facts of D are *true* in M.

3. There is no other interpretation $N = (\Psi, \Sigma')$ such that N satisfies the conditions (1.) and (2.) and Σ'_N is a subsequence[2] of Σ_N. $\square$

---

[2] Given a sequence $X = x_1, \ldots, x_m$, another sequence $Z = z_1, \ldots, z_n$ is a subsequence of X if there exists a strictly increasing sequence $i_1, \ldots, i_k$ of indices of X such that for all j = 1,2 ..., k, we have $x_{i_j} = z_j$.

A domain description D is said to be *consistent* if it has a model.

The query language associated with $\mathcal{L}_0$ denoted by $\mathcal{L}_0^Q$ will consists of facts of $\mathcal{L}_0$ and statements of the following form (called *hypothesis*):

f **after** $a_1, \ldots, a_n$ **at** s

We say the above hypothesis is true in an interpretation $M = (\Psi, \Sigma)$ of a domain description D, if f is true in $\Psi(\Sigma(s) o a_1, \ldots, a_n)$.

Definition 2.5 A domain description D entails a query q (written as $D \models q$) iff q is *true* in all models of D. □

Using the entailment relation we can now formally define planning.

Definition 2.6 Given a goal (a set of fluent literals) G and a domain description D, we say the sequence of actions α is a plan for G w.r.t. D iff for all f in G $D \models f$ **after** αats_N.
□

It is easy to see that this notion planning can be easily extended with the extension of the language.

3 Systematic approach to robot control

In this section we describe some of the extensions to $\mathcal{L}_0$ that are necessary to express the interaction of an autonomous agent with the real world. The semantics of the extended language is based on the semantics of $\mathcal{L}_0$ and will be elaborated in detail in the full version of the paper. One important point is that we need to introduce actions a^- and a^* for every action a. The action a^* refers to the partial execution of a, while the action a^- corresponds to the notion that the action a fails during its execution. Following are some of the extensions with their motivations and their intuitive meaning:

• In the real world we need actions (or behaviors) that guarantee that certain fluent is true during the execution of that action. For example we may have an action *avoid_obstacle* that guarantees that during the execution of this action the fluent *hit_obstacle* is false. Effect of such actions are represented by a *causal maintainance law*, an expression of the form

$$a \ \textbf{ensures} \ f \ \textbf{if} \ p_1, \ldots, p_n \tag{6}$$

where a is an action, and $f, p_1, \ldots, p_n$ $(n \geq 0)$ are fluent literals. $p_1, \ldots, p_n$ are called *preconditions* of (6). We will read this law as "f is guaranteed to be true during the execution of an action a if a is executed in any state of the world in which $p_1 \ldots p_n$ are true".

If $n = 0$, we write the effect law as

$$a \ \textbf{ensures} \ f \tag{6a}$$

• To reason about the value of certain fluents during the execution of an action we allow hypothesis (query) of the following form:

$$f \ \textbf{during} \ a_n \ \textbf{after} \ a_1, \ldots, a_{n-1} \ \textbf{at} \ S \tag{7}$$

Intuitively the above means that "the fluent f will be true during the execution of action a_n in the situation obtained by executing the sequence of actions $a_1, \ldots, a_{n-1}$ in the situation S.

- To be able to reason with actions that may fail during their execution we allow the following law and facts.

$$a \textbf{ affects } f \textbf{ if } p_1, \ldots, p_n \tag{8}$$

$$a \textbf{ executed\_at } S \textbf{ fails} \tag{9}$$

$$a \textbf{ executed\_at } S \textbf{ fails with\_observation } f_1, \ldots, f_n \tag{10}$$

Intuitively, the law (8) means that if $p_1, \ldots p_n$ are true in a state then the action a will affect the fluent f during its execution. Hence, if the action a fails while being executed then the value of f in the resultant state will be unknown.

Intuitively, the fact (9) means that the action a executed in situation S fails and the agent is able to recognize this and return back to its state S. This may happen when the robot is unable to push an heavy object. After the action 'push' fails the robot is in its original state. But certain actions after the failure of its execution the robot may not be able to get back to its earlier position. For example, a robot asked to go from one point to another point in a corridor may find its path blocked halfway and may not be able to go back to its earlier position. In that case the after its fails to execute the action the robot sense the world around it and observes that fluents $f_1, \ldots, f_n$ are true and records an observation of the form (10). It will then start over with the information that the value of the fluents $affected$ by a may be unknown (except that $f_1, \ldots, f_n$ are true) and the values of fluents at S that are unaffected by a remain unchanged.

- Since robots may not always exactly perform that action they are asked to, and moreover since we may not have an exact description of the world it is possible that the robot may be in a different situation than it thinks it is. To overcome this the robot needs to verify its position at intervals. We allow a special action called a_v which does this verification. But the robot can only verify in only certain states. The following expression states when the action a_v is executable.

$$\textbf{possible } a_v \textbf{ if } p_1, \ldots, p_n \tag{11}$$

By executing the action a_v the robot observes certain fluents to be true. This is expressed by a fact of the following form.

$$a_v \textbf{ occurs\_at } S \textbf{ with\_observation } f_1, \ldots, f_n \tag{12}$$

Intuitively, if $f_1, \ldots, f_n$ agree with S then the robot proceeds from S. Otherwise the robot realizes its mistake and starts over with the information that only $f_1, \ldots, f_n$ are true in the current state.

- To make conditional plans the robot has actions that check if a certain fluent is true or not. For a fluent f the corresponding action is denoted by a_f. But the robot can not always perform these actions. For example it can not look in the refrigerator to find out if there is milk unless it is in front of the refrigerator. The following proposition expresses when an action to determine the value of a fluent f can be executed.

$$a_f \textbf{ determines } f \textbf{ if } p_1, \ldots, p_n \tag{13}$$

Intuitively, the above proposition means that the determination action a_f can only be executed in situations where $p_1, \ldots, p_n$ is true.

If a_f is executable in a situation S, then after executing an action a_f a fact of the following form is added to the domain description.

$$a_f \textbf{ occurs\_at } S \textbf{ with\_value } g \tag{14}$$

In the above g is wither f or $\neg f$. Intuitively, the above proposition means that a_f is executable in situation S and g is determined to be true in S.

These are some of the extensions necessary to systematically express information about the interaction between a robot and a real world. In the full paper we will follow the approach in [GL93, BGP95] and discuss how domain descriptions in this extended language can be translated into a logical language.

References

[Bar95] C. Baral. Reasoning about Actions : Non-deterministic effects, Constraints and Qualification. In *IJCAI 95*, pages 2017–2023, 95.

[BG93] C. Baral and M. Gelfond. Representing concurrent actions in extended logic programming. In *Proc. of 13th International Joint Conference on Artificial Intelligence, Chambery, France*, pages 866–871, 1993.

[BGP95] C. Baral, M. Gelfond, and A. Provetti. Representing Actions I: Laws, Observations and Hypothesis. In *Proc. of AAAI 95 Spring Symposium on Extending Theories of Action: Formal theory and practical applications*, http://cs.utep.edu/chitta/publications.html, 1995.

[Bro87] F. Brown, editor. *Proceedings of the 1987 workshop on The Frame Problem in AI.* Morgan Kaufmann, CA, USA, 1987.

[FN71] R. Fikes and N. Nilson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3–4):189–208, 1971.

[GL93] M. Gelfond and V. Lifschitz. Representing actions and change by logic programs. *Journal of Logic Programming*, 17(2,3,4):301–323, 1993.

[GL95] E. Giunchiglia and V. Lifschitz. Dependent fluents. In *IJCAI 95*, pages 1964–1969, 95.

[Kar93] G. Kartha. Soundness and completeness theorems for three formalizations of action. In *Working Papers of the Second Int'l Symp. on Logical Formalizations of Commonsense Knowledge*, 1993.

[KL94] G. Kartha and V. Lifschitz. Actions with indirect effects: Preliminary report. In *KR 94*, pages 341–350, 1994.

[Lin95] F. Lin. Embracing causality in specifying the indirect effects of actions. In *IJCAI 95*, pages 1985–1993, 95.

[LLL+94] Y. Lesperance, H. Levesque, F. Lin, D. Marcu, R. Reiter, and R. Scherl. A logical approach to high level robot programming – a progress report. In *Working notes of the 1994 AAAI fall symposium on Control of the Physical World by Intelligent Systems (to appear), New Orleans, LA*, November 1994.

[LR94] F. Lin and R. Reiter. State constraints revisited. *Journal of Logic and Computation*, 4(5):655–678, October 1994.

[LS92] F. Lin and Y. Shoham. Concurrent actions in the situation calculus. In *Proc. of AAAI-92*, pages 590–595, 1992.

[Mae91] P. Maes, editor. *Designing Autonomous Agents.* MIT/Elsevier, 1991.

[MS94] R. Miller and M. Shanahan. Narratives in the situation calculus. *Journal of Logic and Computation*, 4(5):513–530, october 1994.

[MT95] M. McCain and M. Turner. A causal theory of ramifications and qualifications. In *IJCAI 95*, pages 1978–1984, 95.

[Ped89] E. Pednault. ADL: Exploring the middle ground between STRIPS and the situation calculus. In R. Brachman, H. Levesque, and R. Reiter, editors, *Proc. of the First Int'l Conf. on Principles of Knowledge Representation and Reasoning*, pages 324–332, 1989.

[Pin94] J. Pinto. *Temporal Reasoning in the Situation Calculus.* PhD thesis, University of Toronto, Department of Computer Science, February 1994. KRR-TR-94-1.

[PR93] J. Pinto and R. Reiter. Temporal reasoning in logic programming: A case for the situation calculus. In *Proceedings of 10th International Conference in Logic Programming, Hungary*, pages 203–221, 1993.

[Rei91] R. Reiter. The frame problem in the situation calculus:a simple solution (sometimes) and a completeness result for goal regression. In V. Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation*, pages 359–380. Academic Press, 1991.

[San92] E. Sandewall. Features and fluents: A systematic approach to the representation of knowledge about dynamical systems. Technical report, Institutionen for datavetenskap, Universitetet och Tekniska hogskolan i Linkoping, Sweeden, 1992.

[San94] E. Sandewall. The range of applicability of some non-monotonic logics for strict inertia. *Journal of Logic and Computation*, 4(5):581–616, october 1994.

[SL93] R. Scherl and H. Levesque. The frame problem and knowledge producing actions. In *AAAI 93*, pages 689–695, 1993.

Visually Servoed Manipulation

Using An Active Camera

Bradley J. Nelson
Dept. of Mechanical Engineering (M/C 251)
University of Illinois at Chicago
842 West Taylor Street
Chicago, Illinois 60607-7022

Pradeep K. Khosla
Dept. of Electrical and Computer Engineering
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, Pennsylvania 15213-3890

Abstract

Visual servoing is a robust technique for aligning both static and moving parts using imprecisely calibrated camera-lens-manipulator systems. An important limitation of these systems is the workspace within which the alignment task can be successfully performed due to the position and orientation of the camera. An active camera can extend this region, however this changes the visual representation of the task itself. Therefore, the reference input that drives the visually servoed manipulator must change appropriately. In this paper, a framework that allows for camera-lens motion during visually servoed manipulation is described. The main components of the framework include object schemas and port-based agents. Object schemas represent the task internally in terms of geometric models with attached sensor mappings. Object schemas are dynamically updated by sensor feedback, and thus provide an ability to perform three dimensional spatial reasoning during task execution, a capability traditional image-based visual servoing lacks. Object schemas are also able to dynamically create desired visual representations of the task from which reference inputs for vision-based control strategies are derived. The sensor mappings of object schemas are also used to guide camera motion based on task characteristics. Port-based agents are the executors of the visual reference inputs and the camera motion commands. They interact with the real world through visual servoing control laws. Experimental results that demonstrate system capabilities and performance are presented.

1. Introduction

Visual servoing is a robust technique for guiding a manipulator through an uncertain environment using poorly calibrated camera-lens-manipulator systems. The technique employs a differential view of the world in which only small scene changes between image frames are assumed and compensated for by any of a number of proposed control techniques. Most proposed visual servoing control strategies use image-based techniques [13], because these techniques avoid the need to perform an explicit inverse perspective mapping. In its basic form, image-based visual servoing is a purely reactive manipulator motion strategy. The objective of the control system is simply to make some set of measured feature states defined in the camera's sensor space match some set of desired states defined in the same space by moving the manipulator.

An important limitation to which all visual servoing strategies are subject, is the limited spatial resolution, depth-of-field, and field-of-view inherent to camera-lens systems. Visual servoing systems fail when objects being visually tracked exceed the depth-of-field and field-of-view limitations imposed by vision sensors. These systems also fail when the visual feedback provided by a camera-lens system provides insufficient spatial resolution for visually guiding the manipulation task. A dynamically reconfigurable camera-lens system is able to extend valid tracking regions of objects, however camera-lens motion changes the visual representation of the task. Therefore, image-based reference inputs that drive the visually servoed manipulator performing the task must change as sensor parameters change.

In this paper, we describe and experimentally verify a framework for visually servoed manipulation that integrates visually servoed manipulation with an active camera-lens system via a 3D task representation. This framework provides appropriate image-based reference inputs to a manipulator performing a visually servoed manipulation task during camera-lens motion. Another benefit of the framework is its ability to provide a reasoning capability for geometric motion planners. Within the framework, the task is represented by the actions and interactions of dynamic 3D geometric models of objects in the environment. Each model is augmented by sensor mappings representing the visual sensors used to provide feedback for task execution. These augmented environment models are referred to as *object schemas*, because they simultaneously represent the task and provide an understanding of how the environment is perceived [8]. The mappings are used in two ways. First, they are used to predict how the sensors will view the environment model while the given task is being executed. This provides reference inputs for visual servoing controllers which then guide task execution. Object schemas also use their sensor mappings to determine appropriate motion for active camera-lens systems in order to avoid many of the limitations inherent to static camera-lens system. Object schemas drive both visually servoed manipulators and visually servoed active camera-lens systems. These two visual servoing systems are treated as independent, autonomous visual servoing systems and are referred to as *port-based agents*.

We begin by discussing our task representation in terms of object schemas. Next, visually servoed port-based agents for object manipulation and dynamic sensor reconfiguration are presented. A description of the overall system framework that combines object schemas, port-based agents, and the real world then follows. Finally, we present experimental results which demonstrate the framework operating within a dynamically varying, imprecisely calibrated environment.

2. A Task Representation Using Object Schemas

A key component of our system framework is the structure and content of the task representation. In order to provide the system with the capability to reason about manipulation strategies, the internal representation must contain three dimensional geometric knowledge of the objects being manipulated. The internal representation must also be capable of being correlated with visual information in order to direct the visually servoed manipulator to perform the task. This implies that a visual representation of the task must also exist internally.

The task representation is defined in terms of for objects. The object representation we call an *object schema*. Our object schema definition includes a geometric environment model of an object augmented by sensor mappings that describe how the object will be perceived by the actual system sensors. Forward projective sensor mappings and their associated Jacobian matrix for each visual sensor that exists in the system provides this representation. For a single camera these forward mappings are simply perspective projection mappings and are of the form

$$x_S = \frac{fX_C}{s_x Z_C} \qquad y_S = \frac{fY_C}{s_y Z_C} \tag{1}$$

where x_S and y_S are the projected image coordinates of a point on the object in the internal representation located with respect to the camera frame at (X_C, Y_C, Z_C), f is the focal length of the camera lens, and s_x and s_y are the pixel dimensions of the CCD array. The Jacobian of this mapping is of the form

$$\delta \mathbf{x}_S = \mathbf{J}(\phi)\delta \mathbf{X}_T \tag{2}$$

where $\delta \mathbf{x}_S$ is an infinitesimal displacement vector in sensor space and $\delta \mathbf{X}_T$ is an infinitesimal displacement vector in task space. $\mathbf{J}(\phi)$ is the Jacobian matrix and is a function of the extrinsic and intrinsic parameters of the visual sensor as well as the number of features used for tracking the object and their locations on the image plane.

For the experimental results to be presented, an orthogonal stereo pair is used. Figure 1 shows the coordinate frame definitions for this type of camera-lens configuration. If the axes are aligned as shown in the figure, the Jacobian mapping from task space to sensor space for a single feature can be written as

$$\mathbf{J} = \begin{bmatrix} \frac{f}{s_x Z_{Cl}} & 0 & -\frac{x_{Sl}}{Z_{Cl}} & -\frac{x_{Sl} Y_T}{Z_{Cl}} & \left[\frac{fZ_T}{s_x Z_{Cl}} + \frac{x_{Sl} X_T}{Z_{Cl}}\right] & -\frac{fY_T}{s_x Z_{Cl}} \\ 0 & \frac{f}{s_y Z_{Cl}} & -\frac{y_{Sl}}{Z_{Cl}} & -\left[\frac{fZ_T}{s_y Z_{Cl}} + \frac{y_{Sl} Y_T}{Z_{Cl}}\right] & \frac{y_{Sl} X_T}{Z_{Cl}} & \frac{fX_T}{s_y Z_{Cl}} \\ \frac{x_{Sr}}{Z_{Cr}} & 0 & \frac{f}{s_x Z_{Cr}} & \frac{fY_T}{s_x Z_{Cr}} & \frac{x_{Sr} Z_T}{Z_{Cr}} - \frac{fX_T}{s_x Z_{Cr}} & \frac{x_{Sr} Y_T}{Z_{Cr}} \\ \frac{y_{Sr}}{Z_{Cr}} & \frac{f}{s_y Z_{Cr}} & 0 & -\frac{fZ_T}{s_y Z_{Cr}} & \frac{y_{Sr} Z_T}{Z_{Cr}} & \frac{y_{Sr} Y_T}{Z_{Cr}} - \frac{fX_T}{s_y Z_{Cr}} \end{bmatrix} \tag{3}$$

In (3), we assume the camera-lens parameters are identical for both cameras. The other terms in (3) correspond to Figure 1. The tracking of features will be described in Section 3.1.

Our previous work in defining the sensor placement measure *resolvability* [7] uses the Jacobian mapping to compare various visual sensor systems from a control standpoint in terms of the accuracy of control. This measure quantifies the ability of various monocular and stereo camera configurations to resolve the position and orientation of visually servoed

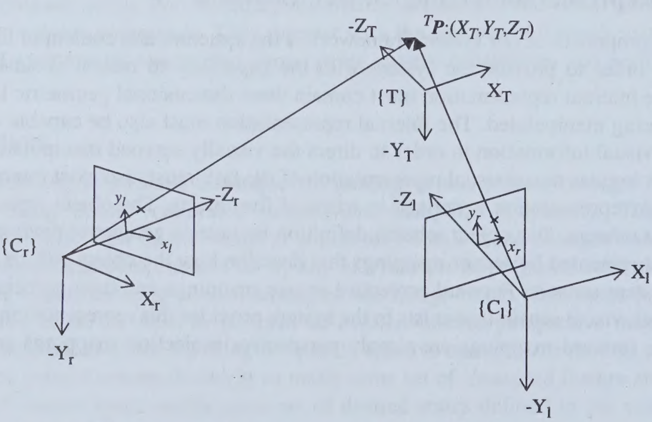


Figure 1. Task frame-camera frame definitions.

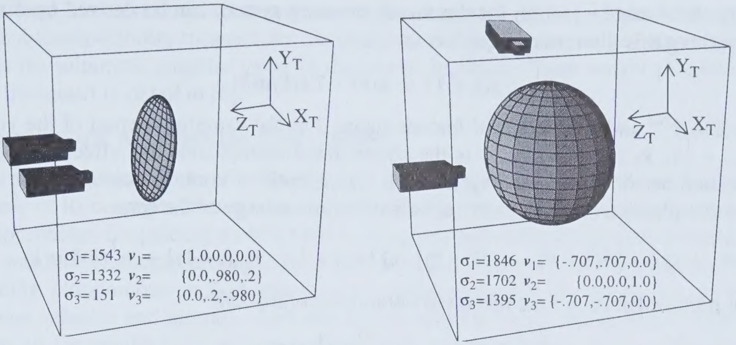

Y_T
Z_T
X_T
σ_1=1543 v_1= {1.0,0.0,0.0}
σ_2=1332 v_2= {0.0,.980,.2}
σ_3= 151 v_3= {0.0,.2,-.980}
Y_T
Z_T
X_T
σ_1=1846 v_1= {-.707,.707,0.0}
σ_2=1702 v_2= {0.0,0.0,1.0}
σ_3=1395 v_3= {-.707,-.707,0.0}

Figure 2. Resolvability ellipsoids for a stereo pair with parallel optical axes and a stereo pair with perpendicular optical axes. The singularvalues of J and the eigenvectors of $J^T J$ are given for each configuration.

objects. As discussed in [7], the measure is easily extendable to other visual sensors including multi-baseline stereo and laser rangefinders. A key component of this measure, the Jacobian mapping from task space to sensor space, is also a critical component of our visual servoing control strategy.

Figure 2 shows the resolvability of two different stereo camera configurations. A sensitivity analysis of resolvability with respect to the variable extrinsic and intrinsic camera-lens parameters shows how camera-lens systems can be reconfigured in order to improve the precision with which objects can be observed and manipulated [6].

Figure 3 shows a diagram of an object schema, including the current internal pose of the geometric model; its desired pose (for controllable schemas) as determined by a meta-schema (an assembly planner or supervisor); and sensor mappings used to direct variable sensor parameters as well as update the current internal pose of the schema based on actual sensor feedback. Note that the pseudoinverse of the Jacobian mapping used for determining sensor resolvability is also used to "servo" the geometric model in order to reduce errors between the current internal visual representation of the object and the actual visual representation.

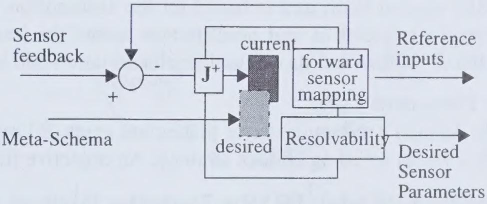

Sensor feedback
+
-
J^+
current
forward sensor mapping
Reference inputs
Meta-Schema
desired
Resolvability
Desired Sensor Parameters

Figure 3. An object schema diagram.

3. Visually Servoed Port-Based Agents

3.1. A Manipulation Agent

The desired visual representation of a task is derived from object schemas. This desired visual representation provides reference inputs to a visual servoing system. The visual servoing system we consider to be an agent that continuously accepts reference commands from object schemas and attempts to achieve the desired visual representation of the task based on these inputs using visual servoing techniques.

Controlled active vision [9] is used to derive a control strategy for visually servoed port-based agents. A state equation for the visual servoing system can be derived by discretizing (2) and writing this discretized equation as

$$\mathbf{x}(k+1) = \mathbf{x}(k) + T\mathbf{J}(k)\mathbf{u}(k) \tag{4}$$

where $\mathbf{x}(k) \in R^{2M}$ and is a vector of feature states, T is the sampling period of the vision system, $\mathbf{u}(k) = \left[\dot{X}_T \; \dot{Y}_T \; \dot{Z}_T \; \omega_{X_T} \; \omega_{Y_T} \; \omega_{Z_T} \right]$ is the commanded manipulator end-effector velocity, and M is the number of features being tracked. Using optimal control techniques, an objective function that places a cost on feature error and control energy of the form

$$F(k+1) = [\mathbf{x}(k+1) - \mathbf{x}_D(k+1)]^{\mathrm{T}} \mathbf{Q} [\mathbf{x}(k+1) - \mathbf{x}_D(k+1)] + \mathbf{u}^{\mathrm{T}}(k)\mathbf{L}\mathbf{u}(k) \tag{5}$$

can be minimized at each time instant to obtain the control law

$$\mathbf{u}(k) = -(T^2\mathbf{J}^{\mathrm{T}}(k)\mathbf{Q}\mathbf{J}(k) + \mathbf{L})^{-1} T\mathbf{J}^{\mathrm{T}}(k)\mathbf{Q}[\mathbf{x}(k) - \mathbf{x}_D(k+1)] \tag{6}$$

The vector $\mathbf{x}_D(k+1)$ represents the desired feature state, i.e. the desired visual representation of the scene, at the next time instant. **Q** and **L** are weighting matrices and allow the user to place a varying emphasis on the feature error and the control input. Extensions to this control strategy and guidelines for choosing the matrices **Q** and **L** can be found in [10]. New sensor configurations can be quickly and easily added to the system by substituting the correct **J** and by adjusting **Q** and **L** accordingly.

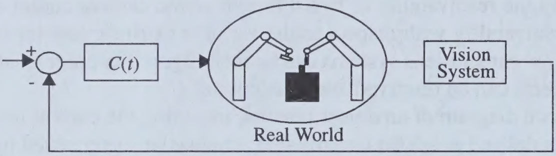


Figure 4. Visual servoing feedback loop.

The measurement of the motion of the features on the image plane must be done continuously and quickly. Our method for measuring this motion is based on optical flow techniques and is a modification of the method proposed in [1]. This technique is known as Sum-of-Squared-Differences (SSD) optical flow, and is based on the assumption that the intensities around a feature point remain constant as that point moves across the image plane. A more complete description of the algorithm and its original implementation can be found in [10].

3.2. A Dynamic Sensor Placement Agent

We have previously defined a dynamic sensor placement agent [6] by integrating sensor placement measures with a visual servoing control strategy. An objective function of the form

$$\begin{aligned} F(k+1) = {} & [\mathbf{x}(k+1) - \mathbf{x}_D(k+1)]^{\mathrm{T}} \mathbf{Q} [\mathbf{x}(k+1) - \mathbf{x}_D(k+1)] + \\ & + \mathbf{u}^{\mathrm{T}}(k)\mathbf{L}\mathbf{u}(k) + \frac{S}{w'(\mathbf{q}(k))} + \frac{U}{(foc)} + \frac{V}{\sigma_{min}} + W(fov) \end{aligned} \tag{7}$$

is minimized at each time instant in order to derive a completely closed loop control law for the dynamic sensor placement agent of the form

$$\begin{aligned} \mathbf{u}(k) = {} & -(\mathbf{J}^{\mathrm{T}}(k)\mathbf{Q}\mathbf{J}(k) + \mathbf{L})^{-1} \mathbf{J}^{\mathrm{T}}(k)\mathbf{Q}\Bigg[(\mathbf{x}(k) - \mathbf{x}_D(k+1)) - \frac{S}{2w'(q)^2}\nabla^{\mathrm{T}}_{\mathbf{u}(k)} w'(\mathbf{q}) \\ & - \frac{U}{2(foc)^2}\nabla^{\mathrm{T}}_{\mathbf{u}(k)}(foc) - \frac{V}{2\sigma_{min}^2}\nabla^{\mathrm{T}}_{\mathbf{u}(k)}\sigma_{min} + \frac{W}{2}\nabla^{\mathrm{T}}_{\mathbf{u}(k)}(fov) \Bigg] \end{aligned} \tag{8}$$

The term *foc* represents a focus measure; *fov* represents a field of view measure; $w(\mathbf{q})$ represents a manipulability measure for the manipulator holding the active camera; and σ_{min} represents the minimum singular value of the image Jacobian. These sensor placement measures are discussed in detail in [6].

4. A Framework of Schemas and Agents

Our system framework of schemas and port-based agents for robotic manipulation draws many of its concepts from past work in the development of expectation-based and verification approaches for guiding mobile robots. An expectation-based approach to scene understanding was first explicitly proposed by Dickmanns [2]. His work was originally concerned with guiding autonomous mobile systems in rapidly changing environments, particularly autonomous vehicles and aircraft. Roth and Jain propose a "verification-based" approach to navigation in the world [11]. A key point of both the expectation and verification-based approaches is that strong internal models of the recent world state are maintained. This significantly reduces the number of hypotheses that must be considered when determining the current state of the world based on sensory data. Neisser's view of the human "perceptual cycle" [8], as Jain points out [5], is similar in many ways to a verification or expectation-based approach. Figure 5 shows a modified representation of Neisser's "perceptual cycle." This figure illustrates our view of the relationship between object schemas of the world, the real world, and where visually servoed agents exist within this scheme. The counter-clockwise flow of information represents the cyclical nature of the system; sensory data updates schemas, which in turn provide inputs to visually servoed agents, which provide sensory data obtained from the real world to the environment model and the object schemas existing within the environment model. This cycle illustrates the interaction between perception of the world, actions taken within this world, and plans made about the world.

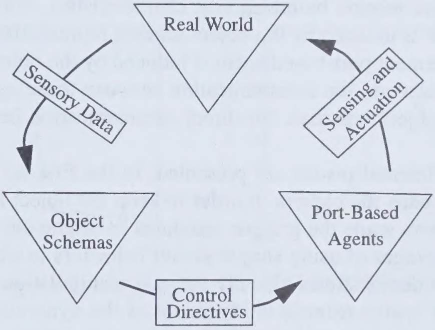


Figure 5. A modified "perceptual cycle" for visually servoed manipulators.

Our proposed integration of visually servoed agents into systems capable of performing manipulation tasks in the real world uses this cycle to clearly delineate the flow of information from the schemas to the visually servoed port-based agents. By clearly separating the system in this way, different types of sensor-based manipulation strategies can be more easily integrated into the system. This is because the internal representation of the task becomes clearly separate from the hardware and low-level behavior-based control systems which perform actions and sensing in the real world.

It is within the environment modeler that the object schemas exist. "Goal achieving functions" provided by a supervisor or some type of motion planner operate on the object schemas in order to describe the task. For example, a supervisor could use a graphical representation of the internal schema representation rather than live video imagery to guide his or her actions. This is similar to the remote teleoperative techniques described in [3] and [4].

5. Hardware Implementation

The schema-agent framework has been implemented on a robotic assembly system consisting of three Puma 560's called the Troikabot. The Pumas are controlled using the Chimera 3.0 reconfigurable real-time operating system[12]. An Adept robot is also used for providing accurate target motion for experimental purposes.

A Datacube Maxtower Vision System calculates the optical flow of the features using the SSD algorithm discussed in Section 3.1. An image can be grabbed and displacements for up to five 16x16 features in the scene can be determined at 30Hz. Stereo system implementations result in half the sampling frequency because only a single digitizer exists on the Datacube. For the stereo system with which experimental results were obtained, visual servoing is performed at 15Hz while tracking ten features. The vision system VME communicates with the robot controller VME using BIT3 VME-to-VME adapters.

The dynamic environment model runs on a Silicon Graphics Crimson. The modeler communicates with the VME-based robot controllers via a Bit3 VME-VME hardware adapter. This allows direct shared memory communication between the motion planner and the manipulators running under the Chimera real-time operating system.

6. Experimental Results

This section demonstrates the use of a moving camera for extending the workspace in which objects can be grasped. Two object schemas are used to represent the gripper and the object to be grasped. A visually servoed port-based agent attempts to achieve the desired visual representation of the scene, while a dynamic sensor placement port-based agent determines appropriate camera motion based on task characteristics. The motion of the visually servoed port-based agent is induced by the object schema representing the gripper. Motion of the dynamic sensor placement port-based agent is induced by the object to be grasped. As dictated by the system framework, the communication between these agents occurs through the task as represented by object schemas. No direct communication between the agents themselves occurs.

Two sets of experimental results are presented. In the first set of results, the dynamic sensor placement agent pans the camera in order to keep the object being tracked within the field-of-view of the camera while the gripper maintains its alignment with the moving object. This illustrates the advantages of using simple sensor behaviors to extend the tracking region of the system. This also demonstrates visually servoed manipulation with a visually servoed camera. The next set of results extends the behavior of the dynamic sensor placement port-based agent to simultaneously maintain the field-of-view behavior by panning and translating along the optical axis of the camera. Vision resolvability is also incorporated as a sensor behavior in order to improve the resolution with which the object is observed and to guarantee convergence of the sensor placement port-based agent.

6.1. Maintaining the Field-of-View Measure During Grasp Alignment

Visually guided manipulation obviously fails when the projection of objects being tracked does not remain on the image plane. Simply by panning a camera about its vertical axis, the visible workspace can be significantly extended. Figure 6 compares the tracking path for visually servoed manipulation that uses two statically mounted cameras with a larger path that is made possible by panning one of the cameras during the alignment task. The controllable camera views the objects's path from a location in the -Z+X quadrant of the graphs shown in Figure 6. The most obvious benefit of allowing the camera to pan is that objects are suc-

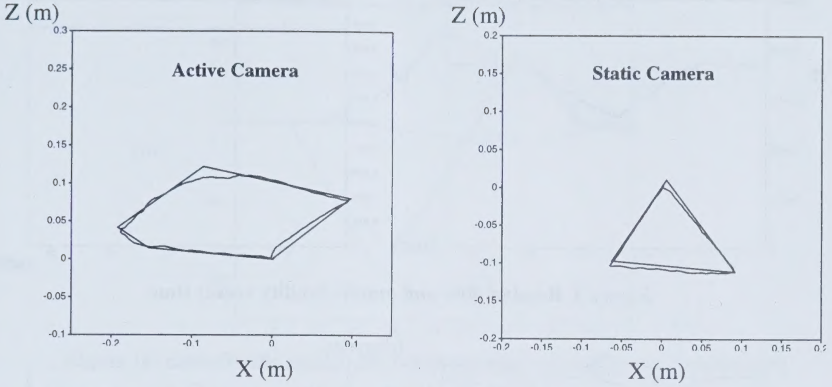


Figure 6. Tracking paths

cessfully tracked for twice the distance along the X axis as they are when the camera remains stationary. From Figure 6, one can observe that the gripper moves approximately 30cm along the X axis. For the static camera case, the maximum possible translational motion along X is approximately 15cm.

The motion of the camera is controlled by the dynamic sensor placement agent and is induced as features on the object and the gripper approach the boundary of the image plane. As the gripper and the object translate along X, the field-of-view measure rises as measured features approach the boundaries of the image plane. This is shown in the left plot of Figure 7.

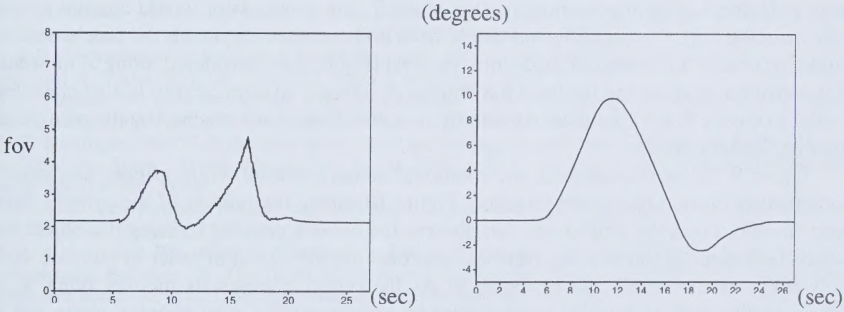


Figure 7. Field-of-view measure and panning angle

This rise in the value of *fov* induces camera motion about Y, and the camera pans with the object and gripper. Figure 8 shows that sensor resolvability is decreased as the depth of the object increases; however, kinematic manipulability remains constant because motion about the panning axis has no effect on singularities. Although this particular configuration for the dynamic sensor placement port-based agent allows no degree of freedom for increasing the resolvability of the vision sensor, the simple field-of view behavior is successful in extending the usable workspace of the system.

6.2. Maximizing Vision Resolvability During Grasp Alignment

In this experiment, camera motion along Z, the optical axis, is allowed in addition to

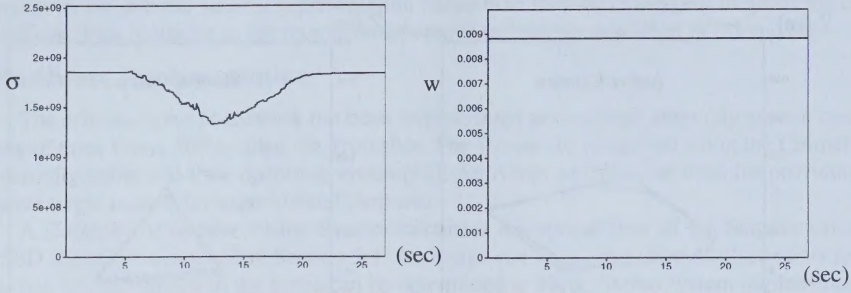


Figure 8. Resolvability and manipulability versus time

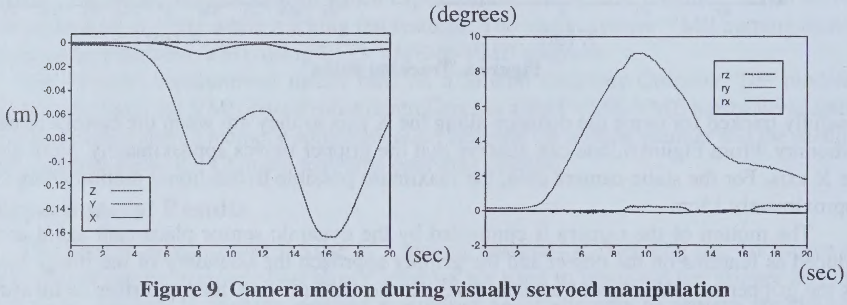


Figure 9. Camera motion during visually servoed manipulation

motion about Y in order to minimize the field-of-view measure. If field-of-view is the only sensor placement criteria governing motion along Z, the manipulator would attempt to move as far from the object as possible and would have no impetus to approach the task. Therefore, in order to ensure a convergent algorithm, resolvability is also considered along Z in order to place a cost on straying too far from the objects of interest. Manipulability is also considered in order to ensure that a kinematic singularity or a joint limit is not reached by the manipulator providing camera motion.

Figure 9 shows translational and rotational camera motion while gripper alignment is maintained with the object being tracked. Figure 10 shows the motion of the gripper during object motion. From the graphs one can observe the camera panning to bring the object back towards the center of the field-of-view and approaching the object in order to increase vision resolvability as the object moves along X. As the object reverses its motion along X, the increase in the field-of-view measure cannot be compensated for by panning alone and the camera begins to move away from the object. Eventually, the field- of-view measure decreases sufficiently to allow vision resolvability to drive the camera back toward the object being observed. The objective of the dynamic sensor placement port-based agent is to minimize the field-of-view measure, the inverse of vision resolvability, and the inverse of manipulability.

7. Conclusion

A framework and task representation for integrating visually servoed manipulation strategies with an active camera system has been described and experimental results presented. A key component of the framework is the use of an expectation-based approach in which augmented geometric models called object schemas are used to describe and reason about the environment. Object schemas are continuously updated based on sensor feedback to insure that the current state of the environment model agrees with current sensory input. Visually servoed port-based agents are used to ensure that desired actions described within the environ-

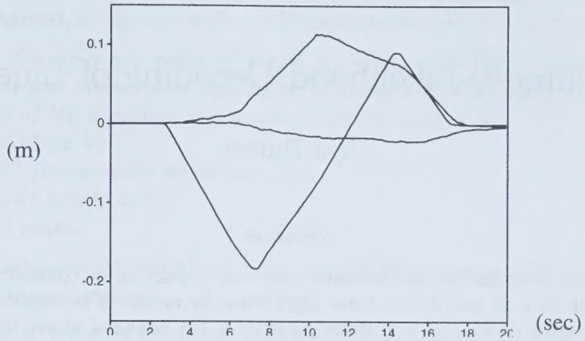


Figure 10. End-effector motion during grasp alignment with a moving camera

ment model are carried out in the real world. These actions include manipulation and dynamic sensor reconfiguration. Our goal is to develop a scalable and extendable framework that will allow the use of plug-and-play types of sensor-based manipulation components.

Acknowledgments

This research was supported in part by the U.S. Army Research Office through Grant Number DAAL03-91-G-0272 and by Sandia National Laboratories through Contract Number AC-3752D.

References

[1] P. Anandan, Measuring visual motion from image sequences. Tech. Rept. COINS-TR-87-21, Amherst, Mass., University of Massachusetts COINS Department, 1987.
[2] E.D. Dickmanns, "Expectation-based Dynamic Scene Understanding," in *Active Vision*, eds. A. Blake and A. Yuille, 303-335, The MIT Press, Cambridge, 1992.
[3] C. Fagerer, D. Dickmanns, and E.D. Dickmanns, "Visual Grasping with Long Delay Time of a Free Floating Object in Orbit," *Autonomous Robots*, 1(1):53-68, 1994.
[4] G. Hirzinger, "ROTEX-the first space robot technology experiment," *Experimental Robotics III: The Third Int. Symp., Kyoto, Japan, Oct. 28-30, 1993*, eds. T. Yoshikawa and F. Miyazaki, Springer-Verlag, pp.579-598, 1994.
[5] R. Jain, "Environment Models and Information Assimilation," Technical Report RJ 6866(65692), IBM-Yorktown Heights, 1989.
[6] B. Nelson and P.K. Khosla, "Integrating Sensor Placement and Visual Tracking Strategies," in *Experimental Robotics III: The Third International Symposium, Kyoto, October 28-30, 1993*, eds. T. Yoshikawa and F. Miyazaki, Springer-Verlag, London, pp. 169-181, 1994.
[7] B.J. Nelson and P.K. Khosla, "The Resolvability Ellipsoid for Visual Servoing," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR94)*, pp. 829-832, 1994.
[8] U. Neisser, *Cognition and Reality*, W.H. Freeman and Co., New York, 1976.
[9] N.P. Papanikolopoulos, Khosla, P.K. and Kanade, T., "Adaptive robotic visual tracking," *Proc. of the American Control Conference*. Evanston, IL.:American Automation Control Council, pp. 962-967, 1991.
[10]N.P. Papanikolopoulos, Nelson, B. and Khosla, P.K., "Full 3-d tracking using the controlled active vision paradigm," *Proc. 1992 IEEE Int. Symp. on Intelligent Control (ISIC-92)*, pp. 267-274, 1992.
[11]Y. Roth and R. Jain, "Integrated Architectures for Autonomous Systems," *Proc. of the SPIE - The Int. Society for Optical Engineering*, vol. 1571, pp. 628-639, 1991.
[12]D.B. Stewart, Schmitz, D.E. and Khosla, P.K., "The Chimera II real-time operating system for advanced sensor-based control systems," *IEEE Trans. Sys., Man Cyber.* 22(6):1282-1295., 1992.
[13]L.E. Weiss, "Dynamic visual servo control of robots: an adaptive image-based approach," Ph.D Thesis CMU-RI-TR-84-16, Pittsburgh, PA:The Robotics Institute Carnegie Mellon University, 1984.

Near-Maximum-Likelihood Decoding of Linear Codes

Ilya Dumer

Abstract

We study near-maximum-likelihood decoding of q-ary linear codes of any length n and code rate R, and design three algorithms by means of incomplete trellises. Decoding error probabilities of these algorithms are bounded above by twice the error probabilty of full maximum-likelihood (ML) decoding, and tend to the latter as n grows. The corresponding decoding complexities are bounded above by the exponential orders $q^{nR(1-R)/(1+R)}$ for virtually all long linear codes, $q^{nR(1-R)}$ for cyclic codes, and $q^{n(1-R)/2}$ for all codes. Examples are considered for short codes.

1 Introduction

Trellis realization of linear block codes was introduced in [1] and [2] (see also [3] for a similar trellis construction). It was proved in [2] that for a linear (q, n, R) code being used over a memoryless channel ML decoding can be executed in $q^{n(c+o(1))}$ operations, where the relative complexity exponent c is bounded above by $\min(R, 1-R)$ and $o(1) \to 0$ as $n \to \infty$. Trellis realization of the best binary concatenated codes meeting the asymptotic Varshamov-Gilbert bound allows to obtain (see [4]) the complexity exponent $c \leq \min(R, H(2\delta) - 1 + R)$ for ML decoding, where $\delta \leq 1/2$ is the root of the equation $\delta = H^{-1}(1-R)$ and $H(x) = -x \log_2 x - (1-x) \log_2(1-x)$ is the binary entropy, $0 < x < 1$. Design of codes with small trellis complexity has recently acquired a great interest after the paper [5] had been published (see [6], [7], [8] and many more references therein). Now trellis complexity is recognized as an important intrinsic parameter of a code. Apart from the upper bound $c \leq \min(R, 1-R)$ on the relative trellis complexity, the corresponding lower bounds were derived in [6], [9], and [10] for all codes.

In this correspondence we use a different approach. Namely, we make use of an *incomplete* trellis decoding, whose paths and states are dynamically constructed depending upon a received output. In this way the relative complexity exponent c is substantially reduced at a negligible cost in terms of decoding error probability. The obtained bounds reduce the upper bound $c \leq \min(R, 1-R)$ in up to three times, and lie below the lower bound [9] for code rates $R \leq 0.3$.

2 Results

Define a memoryless channel by a q-ary input alphabet X, an arbitrary output alphabet Y ($|Y| \leq \infty$), and conditional probability $p(y|x)$, $x \in X$, $y \in Y$. Consider any finite output subset $Y_\alpha \in Y$ and the corresponding matrix $p_\alpha = ||p(y|x)||, x \in X, y \in Y_\alpha$. A channel is called *symmetric* if there exists a partition $Y = \cup_\alpha Y_\alpha$ into disjoint finite subsets Y_α such, that in any matrix p_α any row and column are the permutations of any other row and column respectively. Examples include discrete channels with additive noise, AWGN-channel, or channel with q-PSK modulation and two-dimensional AWGN.

Proposition 1 *Linear q-ary codes of length $n \to \infty$ and code rate $R, 0 < R < 1$, can be decoded in any memoryless symmetric channel with a decoding error probability equivalent to that of ML decoding, and decoding complexity $\kappa = q^{n(c+o(1))}$, where $o(1) \to 0$ and c is bounded above by*
$R(1-R)/(1+R)$ for virtually all linear codes,
$R(1-R)$ for cyclic codes, and
$(1-R)/2$ for all codes.

We also exhibit some numerical examples for AWGN-channels. This discussion leads to the following major problem.

Open Problem 1 *What is an extra complexity cost of the soft decision suboptimal decoding in a AWGN channel as compared to its hard decision counterpart (that is, the minimum distance decoding in a binary symmetric channel) ?*

References

[1] L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory,* vol. IT-20, pp. 284-287, 1974.

[2] J.K. Wolf, "Efficient maximum likelihood decoding of linear codes using a trellis", *IEEE Trans. Inform. Theory,* vol. IT-24, pp. 76-80, 1978.

[3] J.L. Massey, "Foundations and methods of channel coding", *Proc. Int. Conf. Inform. Theory and Systems*, NTG-Fachberichte vol. 65,
pp. 148-157, 1978.

[4] I.I. Dumer, "On the complexity of maximum likelihood decoding of the best concatenated codes", In: *Proc. 8 All-Union Conf. on Coding Theory. Vol. 2*, Moscow-Kuibyshev, p. 66-69, 1981 (in Russian).

[5] G.D. Forney, Jr., "Coset codes II: Binary lattices and related codes", *IEEE Trans. Inform. Theory*, vol. IT-34, pp. 1152-1187, Sept. 1988.

[6] D.J. Muder, "Minimal trellises for block codes", *IEEE Trans. Inform. Theory*, vol. IT-34, pp. 1049-1053, 1988.

[7] R.J. McEliece, "On the BCJR trellis for block codes," *California Inst. of Technology. Preprint,* 49 p., Sept. 1994.

[8] O. Amrani, B. Be'ery, A. Vardy, F.-W. Sun, and H.C.A. van Tilborg, "The Leech lattice and the Golay codes: bounded-distance decoding and multilevel constructions," *IEEE Trans. Inform. Theory,* vol. IT-40, pp. 1030-1043, July 1994.

[9] V.V. Zyablov and V.R. Sidorenko, "Bounds on complexity of trellis decoding of linear codes," *Probl. Peredachi Informatsii*, vol. 29, pp. 3-9, 1993.

[10] A. Lafourcade and A. Vardy, "Asymptotically good codes have infinite trellis complexity," *IEEE Trans. Inform. Theory*, vol. IT-41, pp. 555-559, Mar. 1995.

Heuristic Search Algorithms for Soft-Decision Decoding[1]

Ching-Cheng Shih, Christopher R. Wulff,
Carlos R. P. Hartmann, and Chilukuri K. Mohan
School of Computer and Information Science
Syracuse University, Syracuse, NY 13244-4100, USA
Email: ccshih/crwulff/hartmann/mohan@top.cis.syr.edu

Abstract

In this paper, we present three novel and efficient soft-decision decoding algorithms for linear block codes. The first algorithm MA^* conducts heuristic search through a code tree of the transmitted code when the decoding problem is transformed into a graph-search problem. The second algorithm H^* searches for a minimum cost codeword by optimizing a lower bound on the cost of a minimum cost codeword when the decoding algorithm is transformed into a minimum cost problem among all codewords of the transmitted code. The third algorithm hybridizes MA^* and a modified H^* to take advantage of the strengths of these two algorithms and make the decoding more efficient. Simulation results of this hybrid algorithm are presented for the $(128, 64)$ and the $(256, 131)$ binary extended BCH codes.

1 Introduction

The use of block codes is a well-known error-control technique for reliable transmission of digital information over noisy communication channels. Linear block codes with good coding gains have been known for many years; however, these block codes have not been used in practice due to the unavailability of an efficient soft-decision decoding algorithm.

This paper deals with *maximum-likelihood soft-decision decoding* (MLSD) as well as *sub-optimal soft-decision decoding* (SOSD) of linear block codes. *Maximum-likelihood decoding* (MLD) minimizes the probability of decoding to an incorrect codeword when all codewords have equal probability of being transmitted. *Soft-decision* decoding uses real numbers (e.g., the analog output of filters matched to the signals) associated with all components of the codeword. Soft-decision decoding can provide about 2 dB of additional coding gain when compared to hard-decision decoding. Recently developed decoding algorithms [11, 5, 7, 6] are very efficient for block codes of short-to-moderate lengths, but become impractical for long codes transmitted over channels with low SNR. Efficient decoding algorithms for long linear block codes are desirable; because of noise averaging, we expect codes of greater lengths to be more "effective" than codes of shorter lengths [1, Sec. 1.1].

In this paper, we present three new decoding algorithms for linear block codes. The first algorithm MA^* searches through a code tree of the transmitted code when the decoding problem is converted into a graph-search problem, taking into consideration more properties

---

[1] This work was partially supported by the National Science Foundation under Grant NCR-9205422. C. R. Wulff was supported by a Research Experience for Undergraduates Supplement of Grant NCR-9205422.

of the code, and is considerably more efficient than the original A^* algorithm [6]. The second algorithm H^* searches for a minimum cost codeword by optimizing a lower bound on the cost of a minimum cost codeword when the decoding algorithm is transformed into a minimum cost problem among all codewords of the transmitted code. This algorithm can be modified and incorporated with other decoding algorithms to reduce the search space during the decoding process. The third algorithm hybridizes MA^* and a modified H^* to take advantage of the strengths of these two algorithms and make decoding more efficient.

In Section 2 we review maximum-likelihood decoding of linear block codes. MA^* is described in Section 3 and H^* is described in Section 4. The hybrid algorithm is presented in Section 5. Simulation results for the (128, 64) and the (256, 131) binary extended BCH codes are given in Section 6. Concluding remarks are presented in Section 7.

2 Preliminaries

Let $\boldsymbol{C}$ be a binary (n, k) linear code with generator matrix $\boldsymbol{G}$, and let $\boldsymbol{c} = (c_0, c_1, \ldots, c_{n-1})$ be a codeword of $\boldsymbol{C}$ transmitted over a time-discrete, memoryless channel with output alphabet $\boldsymbol{B}$. Furthermore, let $\boldsymbol{r} = (r_0, r_1, \ldots, r_{n-1})$, $r_j \in \boldsymbol{B}$ denote the received vector, and assume that $\boldsymbol{Pr}(r_j|c_i) > 0$ for $r_j \in \boldsymbol{B}$ and $c_i \in \boldsymbol{GF}(2)$. The MLSD decoding rule for a time-discrete, memoryless channel can be formulated as follows [9]: given a received vector $\boldsymbol{r} = (r_0, r_1, \ldots, r_{n-1})$, find a codeword $\boldsymbol{c} = (c_0, c_1, \ldots, c_{n-1}) \in \boldsymbol{C}$ that minimizes $\sum_{j=0}^{n-1} (\phi_j - (-1)^{c_j})^2$, where $\phi_j = \ln \frac{\boldsymbol{Pr}(r_j|0)}{\boldsymbol{Pr}(r_j|1)}$. We may therefore consider that the "received vector" is $\boldsymbol{\phi} = (\phi_0, \phi_1, \ldots, \phi_{n-1})$.

Simulation results [3, 6] showed that it is very important that we select the "best" set of linearly independent codeword variables, corresponding to the set of the k-most reliable linearly independent positions in the received vector $\boldsymbol{\phi}$, where ϕ_i is said to be more "reliable" than ϕ_j if and only if $\mid \phi_i \mid > \mid \phi_j \mid$. Each time a new vector is received, the k-most reliable, independent positions must be determined anew. For simplicity of notation, we will henceforth take $\phi_0, \phi_1, \ldots, \phi_{k-1}$ to be the set of k-most reliable linearly independent positions with $\mid \phi_j \mid \geq \mid \phi_{j+1} \mid$ for $j = 0, 1, \ldots k-2$. Furthermore, we assume that the first k columns of the generator matrix $\boldsymbol{G}$ form a $k \times k$ identity matrix.

Every codeword of an (n, k) linear code $\boldsymbol{C}$ corresponds to a path through a code tree containing $n+1$ levels. The root is called the *start node*, which is at level -1. There are two branches, labeled 0 and 1, respectively, that leave each node at the first k levels. After the first k levels, there is only one branch leaving each node. The 2^k leaves are called *goal nodes*, which are at level $n-1$. Let $c_0, c_1, \ldots, c_{k-1}$ be the sequence of labels encountered when traversing a path from the start node to a node m at level $k-1$. Then $c_k, c_{k+1}, \ldots, c_{n-1}$, the sequence of labels encountered when traversing a path from node m to a goal node, can be obtained as follows: $(c_0, c_1, \ldots, c_{k-1}, c_k, \ldots, c_{n-1}) = (c_0, c_1, \ldots, c_{k-1})\ \boldsymbol{G}$.

We now specify the arc cost in the code tree of $\boldsymbol{C}$. The arc from a node at level $t-1$ to a node at level t is assigned the cost $(\phi_t - (-1)^{c_t})^2$, where c_t is the label of the arc. Thus the decoding problem is equivalent to finding the "optimal" path from the start node to a goal node which corresponds to a codeword $\boldsymbol{c} = (c_0, c_1, \ldots, c_{n-1})$ such that $\sum_{j=0}^{n-1} (\phi_j - (-1)^{c_j})^2$ is minimum among all paths from the start node to the goal nodes.

3 Modified Algorithm A^* (MA^*)

In this section, we describe MA^*, a new decoding algorithm that searches through a code tree of the transmitted code when the decoding problem is converted into a graph-search problem through a code tree. MA^* takes into consideration more properties of the code and is considerably more efficient than the original A^* algorithm for decoding [8, 6].

MA^* uses a priority-first search strategy to search for an optimal path from the start node to a goal node in a code tree. For a fixed positive integer q, two functions, f_q and low_q, are defined for every node in the code tree to take advantage of the information provided by the received vector and the inherent properties of the transmitted code. In MA^*, search is guided by the evaluation function f_q. The algorithm maintains a list $\mathcal{L}$ of nodes of the code tree that are candidates to be expanded. The algorithm selects for expansion the node in $\mathcal{L}$ with minimum values of function f_q. If it selects a goal node for expansion, it has found an "optimal" path from the start node to a goal node whose labels correspond to a codeword that minimizes the error probability when we assume all codewords have equal probability of being transmitted. Function low_q is a lower bound on the cost of an optimal path that goes through a node in the code tree. For every node m, $low_q(m) \geq f_q(m)$. This algorithm also stores an upper bound UB on the cost of an optimal path. If the value of $low_q(m)$ for a node is greater than or equal to UB, this node can be discarded.

For every node m in the code tree and a fixed positive integer q, functions f_q and low_q are defined as follows: $f_q(m) = g(m) + h_q(m)$ and $low_q(m) = g(m) + h'_q(m)$, where $g(m)$ is the actual cost of the path from the start node s to node m, $h_q(m)$ estimates the cost of the minimal cost path from node m to a goal node, and $h'_q(m)$ estimates the cost of the minimal cost path from node m to a goal node while taking into consideration the parity check property of the code.

For a positive integer q, function h_q takes into consideration q binary n-tuples, which satisfy the Hamming weight constraints, whose corresponding paths from level 0 to level $k-1$ in the code tree go through node m, and which are closer to the received vector $\boldsymbol{\phi}$ than any other binary n-tuple vectors.

Let $HW = \{w_i | 0 \leq i \leq J\}$ be the set of all $J+1$ distinct Hamming weights that codewords of C may have. Furthermore, assume $w_0 < w_1 < \cdots < w_J$. The heuristic function h_q utilizes linearity of C and the fact that the Hamming distance between any two codewords of C must belong to HW. Our new heuristic function h_q is defined as follows, with respect to $\boldsymbol{c}_{seed}$, a codeword called the seed of the decoding algorithm.

1. For nodes at level s, $0 \leq s < k$:

   Let m be a node at level s and let $\overline{v}_0, \overline{v}_1, \ldots, \overline{v}_s$ be the labels of the path $\boldsymbol{P}_m$ from the start node to node m. Let $T(m)$ be the set of all binary n-tuple vectors $\boldsymbol{v}$ whose first $s+1$ entries are the labels of $\boldsymbol{P}_m$ and $d_H(\boldsymbol{v}, \boldsymbol{c}_{seed}) \in HW$, where $d_H(\boldsymbol{x}, \boldsymbol{y})$ is the Hamming distance between $\boldsymbol{x}$ and $\boldsymbol{y}$. That is, $T(m)$=$\{\boldsymbol{v}|\boldsymbol{v}= (\overline{v}_0, \overline{v}_1, \ldots, \overline{v}_s, v_{s+1}, \ldots, v_{n-1})$

and $d_H(\boldsymbol{v}, \boldsymbol{c}_{seed}) \in HW\}$. We will construct a sequence of finite non-empty subsets of $\{0,1\}^n$, $T_1(m) \supseteq T_2(m) \supseteq \ldots \supseteq T_q(m)$, and for every $T_i(m)$ we define $\boldsymbol{v}_i = (\overline{v}_0, \overline{v}_1, \ldots, \overline{v}_s, v_{i(s+1)}, \ldots, v_{i(n-1)}) \in T_i(m)$ as the vector which satisfies

$$\sum_{j=s+1}^{n-1} (\phi_j - (-1)^{v_{ij}})^2 = \min_{\boldsymbol{v} \in T_i(m)} \left\{ \sum_{j=s+1}^{n-1} (\phi_j - (-1)^{v_j})^2 \right\}.$$

For $1 \leq i \leq q$, $T_i(m)$ is now recursively defined as $T_1(m) = T(m)$ and

$$T_{i+1}(m) = \begin{cases} T_i(m) & \text{if } \boldsymbol{v}_i \text{ is a codeword;} \\ T_i(m) - \{\boldsymbol{v}_i\} & \text{otherwise.} \end{cases}$$

Note that $T_i(m) \neq \emptyset$ since $\boldsymbol{u}\ \boldsymbol{G}$ is a codeword in $T(m)$, where $\boldsymbol{u} = (\overline{v}_0, \overline{v}_1, \ldots, \overline{v}_s, 0, \ldots, 0)$ is a binary k-tuple vector. The heuristic function h_q is defined as $h_q(m) = \sum_{j=s+1}^{n-1} (\phi_j - (-1)^{v_{qj}})^2$, where $\boldsymbol{v}_q = (\overline{v}_0, \overline{v}_1, \ldots, \overline{v}_s, v_{q(s+1)}, \ldots, v_{q(n-1)})$.

2. For nodes at level $s, k-1 \leq s < n$:

   Let m be a node at level s, $h_q(m) = \sum_{i=s+1}^{n-1} (\phi_i - (-1)^{v_{qi}})^2$, where $\boldsymbol{v}_q = (\overline{v}_0, \overline{v}_1, \ldots, \overline{v}_s, v_{q(s+1)}, \ldots, v_{q(n-1)}) = (\overline{v}_0, \overline{v}_1, \ldots, \overline{v}_{k-1})\ \boldsymbol{G}$, and $v_{q(s+1)}, v_{q(s+2)}, \ldots, v_{q(n-1)}$ are the labels of the unique path $\boldsymbol{P}_m$ from node m to a goal node.

Function h'_q also takes into consideration the fact that components in the redundancy part of a codeword can be determined by the parity checks. Let $\boldsymbol{H} = [h_{ij}]$ be the parity check matrix of code $\boldsymbol{C}$ in canonical form, i.e., the last $n-k$ columns of $\boldsymbol{H}$ form an identity matrix. Let $S_H = \{\boldsymbol{h}_0, \boldsymbol{h}_1, \ldots, \boldsymbol{h}_{n-k-1} \mid \boldsymbol{h}_i = (h_{i0}, h_{i1}, \ldots, h_{i(n-1)})$ is the i^{th} row of $\boldsymbol{H}\}$ and for a node m at level s, $0 \leq s < k$, such that $\overline{v'}_0, \overline{v'}_1, \ldots, \overline{v'}_s$ are the sequence of labels of the path $\boldsymbol{P}_m$ from the start node to node m, we define $S_H(m) = \{\boldsymbol{h}_i \mid \boldsymbol{h}_i \in S_H,\ h_{ij} = 0$ for $j = s+1, s+2, \ldots, k-1\}$. According to linear property of the code, if $S_H(m) \neq \emptyset$, then for any codeword $\boldsymbol{c} = (c_0, c_1, \ldots, c_{n-1}) \in \boldsymbol{C}$ such that $c_0 = \overline{v'}_0, c_1 = \overline{v'}_1, \ldots, c_s = \overline{v'}_s$, we have $c_{k+i} = (\overline{v'}_0 h_{i0} \oplus \overline{v'}_1 h_{i1} \oplus \ldots \oplus \overline{v'}_s h_{is})$ for all $\boldsymbol{h}_i \in S_H(m)$, where $\oplus$ denotes the modulo 2 addition operator.

Function h'_q is defined analogously to h_q except that it is with respect to $T'(m)$ which is the set of all binary n-tuple vectors $\boldsymbol{v}'$ whose first $s+1$ entries are $\overline{v'}_0, \overline{v'}_1, \ldots, \overline{v'}_s$, whose $(k+i)^{th}$ entry is $(\overline{v'}_0 h_{i0} \oplus \overline{v'}_1 h_{i1} \oplus \ldots \oplus \overline{v'}_s h_{is})$ for all $\boldsymbol{h}_i \in S_H(m)$, and which satisfy the Hamming weight constraints, i.e., $d_H(\boldsymbol{v}', \boldsymbol{c}_{seed}) \in HW$, where $\boldsymbol{c}_{seed}$ is the seed of the decoding algorithm. That is, $T'(m) = \{\boldsymbol{v}' \mid \boldsymbol{v}' = (\overline{v'}_0, \overline{v'}_1, \ldots, \overline{v'}_s, v'_{s+1}, \ldots, v'_{n-1}),\ v'_{k+i} = (\overline{v'}_0 h_{i0} \oplus \overline{v'}_1 h_{i1} \oplus \ldots \oplus \overline{v'}_s h_{is})$ for all $\boldsymbol{h}_i \in S_H(m)$, and $d_H(\boldsymbol{v}, \boldsymbol{c}_{seed}) \in HW\}$. Note that $T'(m) \neq \emptyset$ since the codeword $\boldsymbol{u}\ \boldsymbol{G}$ is also in $T'(m)$, where $\boldsymbol{u} = (\overline{v}_0, \overline{v}_1, \ldots, \overline{v}_s, 0, \ldots, 0)$ is a binary k-tuple. Therefore, if $\boldsymbol{v}'_q = (v'_{q0}, v'_{q1}, \ldots, v'_{q(n-1)})$ is the vector used to calculate $h'_q(m)$, then $v'_{q0} = \overline{v'}_0, v'_{q1} = \overline{v'}_1, \ldots, v'_{qs} = \overline{v'}_s$, and whenever $S_H(m) \neq \emptyset$, then $v'_{q(k+i)} = (\overline{v'}_0 h_{i0} \oplus \overline{v'}_1 h_{i1} \oplus \ldots \oplus \overline{v'}_s h_{is})$ for all $\boldsymbol{h}_i \in S_H(m)$.

If no restriction is placed on the size of list $\boldsymbol{\mathcal{L}}$, then the decoding algorithm based on MA^* is an MLSD decoding algorithm. However, even this new decoding algorithm becomes impractical for codes of long lengths transmitted over channels with low SNR, since the number of nodes stored in list $\mathcal{L}$ is still too large. Simulation results have shown that the number of nodes needed to be stored in list $\mathcal{L}$ before an optimal path is found is considerably smaller than the total number of nodes stored before the algorithm terminates. Therefore, limiting the search does not significantly affect the performance of the algorithm. In our SOSD decoding algorithm, we limit the size of list $\mathcal{L}$ according to the following criterion:

If a node m needs to be stored in list $\mathcal{L}$ when the size of list $\mathcal{L}$ has reached a given upper bound M_B, then we discard the node with larger f_q value between node m and the node with the maximum value of function f_q in list $\mathcal{L}$.

If $M_B \geq (2^{k+1}-1)$, then MA^* is an MLSD decoding algorithm. In this case, the algorithm will still find an optimal path even if in the computation of h_q the algorithm considers all the Hamming weights of any super set of HW. When $M_B < (2^{k+1}-1)$, the performance of the SOSD decoding algorithm using HW is never worse than the performance of the one that uses a superset of HW.

4 New decoding algorithm – H^*

In this section, we transform the decoding problem into a cost minimization problem and describe H^*, a new decoding algorithm which searches for a minimum cost codeword by optimizing a lower bound on the cost of a minimum cost codeword.

The MLSD decoding problem is equivalent to the minimum cost problem among all codewords in C if we define the cost of a codeword as follows. Let $\boldsymbol{c} = (c_0, c_1, \ldots, c_{n-1})$ be a codeword in C. We define the cost of c_j as $(\phi_j - (-1)^{c_j})^2$. The cost of $\boldsymbol{c}$ can now be defined as the sum of the cost for all of its components, i.e., the cost of $\boldsymbol{c}$ is equal to $\sum_{j=0}^{n-1}(\phi_j - (-1)^{c_j})^2$. The minimum cost problem can be formulated as: given $\boldsymbol{\phi}$, find a codeword $\boldsymbol{c} = (c_0, c_1, \ldots, c_{n-1}) \in C$ that minimizes $\sum_{j=0}^{n-1} (\phi_j - (-1)^{c_j})^2$.

In H^*, a lower bound low'_j on the cost of a minimum cost codeword is defined for every index i, $0 \leq i < n$, to estimate the cost of a minimum cost codeword whose i^{th} component has a fixed value, where j is a positive integer. The algorithm also keeps an upper bound UB on the cost of a minimum cost codeword. If $low'_j(i)$ is greater than or equal to UB, then the value of the i^{th} component of a minimum cost codeword can be determined. This process repeats until the first k components of a minimum cost codeword have been determined.

Let $\boldsymbol{c}^* = (c_0^*, c_1^*, \ldots, c_{n-1}^*)$ be a minimum cost codeword. For a positive integer j and every index $i \in \{0, 1, \ldots, n-1\}$ such that the c_i^* has not yet been determined, low'_j is defined as $low'_j(i) = g'(\overline{I}) + h''_j(i)$, where $g'(\overline{I})$ is the sum of the actual cost for all components of a minimum cost codeword whose values have been determined by the algorithm, $\overline{I} \subset \{0, 1, \ldots, n-1\}$ is the set which contains all indices t such that the value of the t^{th} component of a minimum cost codeword has been determined, i.e., c_t^* is already determined, and $h''_j(i)$ estimates the sum of the cost for all undetermined components of a minimum cost codeword whose i^{th} component has a fixed value.

For a set $\overline{I} \in \{0, 1, \ldots, k-1\}$ such that c_t^* has been determined for every $t \in \overline{I}$, $g'(\overline{I}) = \sum_{l \in \overline{I}}(\phi_l - (-1)^{c_i^*})^2$. Let $I = \{0, 1, \ldots, k-1\} - \overline{I}$ be the set of indices i in which c_i^* is still undetermined. Assume that $\boldsymbol{c} = (c_0, c_1, \ldots, c_{n-1}) \in C$ is the codeword whose cost is the upper bound UB and let $\boldsymbol{c}_{seed} \in C$ be the seed of the decoding algorithm. For a positive integer j and every index $i \in I$, function h''_j is defined analogously to the heuristic function h'_j described in Section 3 except that it is with respect to $T''(i)$, the set of all binary n-tuple vectors $\boldsymbol{v}''$ whose i^{th} component is $c_i \oplus 1$ for some $i \in I$, whose l^{th} component is c^*_l for every $l \in \overline{I}$, and which satisfy the Hamming weight constraints. That is, $T''(i) = \{\boldsymbol{v}'' \mid \boldsymbol{v}'' = (v''_0, v''_1, \ldots, v''_{n-1}),\ v''_i = c_i \oplus 1,\ v''_l = c^*_l \text{ for all } l \in \overline{I}, \text{ and } d_H(\boldsymbol{v}'', \boldsymbol{c}_{seed}) \in HW\}$. Note that $T''(i) \neq \emptyset$ since $\boldsymbol{u}'\,\boldsymbol{G}$ is a codeword in $T''(i)$, where $\boldsymbol{u}' = (u'_0, u'_1, \ldots, u'_{k-1})$ is a binary k-tuple vector such that $u'_i = c_i \oplus 1$, $u'_l = c^*_l$ for all $l \in \overline{I}$, and $u''_t = 0$ for all $t \in \{0, 1, \ldots, k-1\} - \{i\} - \overline{I}$. We also note that function h''_j is defined for every index i such that the value for the i^{th} component of a minimum cost codeword is still undetermined, while function h'_j is defined for every node in the code tree.

During the decoding process, the search space shrinks by half when the value of one component of the minimum cost codeword is fixed. We note that the number of vectors needed to be generated before the algorithm finds a minimum cost codeword is of order $O(2^n)$. However, algorithm H^* can be modified and incorporated with other decoding algorithms to reduce the search space during the decoding process and make the decoding more efficient.

5 Hybrid decoding algorithm

In this section, we describe a new decoding algorithm which hybridizes a modified H^* and algorithm MA^*. This hybrid algorithm takes advantage of the strengths of these two algorithms and makes the decoding more efficient.

When the decoding problem is converted into a graph-search problem through a code tree we are interested in finding an optimal path from the start node to a goal node in the code tree of the transmitted code. The corresponding codeword of an optimal path in the code tree is a minimum cost codeword when the decoding problem is transformed into a minimum cost problem among codewords of the transmitted code. In the hybrid algorithm, a modified H^* is first applied to determine the values of some components of the minimum cost codeword. For every index $i \in \{0, 1, \ldots, k-1\}$ such that the value of component i of the minimum cost codeword is still undetermined, the modified H^* generates at most p binary n-tuple vectors in $T''(i)$ to estimate the cost of a minimum cost codeword, where p is a fixed positive integer. If some components of the minimum cost codeword are determined, the code tree shrinks and the search space is reduced. MA^* is then applied to search for an optimal path through this subtree. During the decoding process, the modified H^* is applied again when MA^* finds a better upper bound. This process repeats until either list $\boldsymbol{\mathcal{L}}$ in MA^* is empty or the values of the first k components of a minimum cost codeword have been determined by H^*. For all $i \in I$ such that c^*_i is determined by the modified H^* during this decoding process, $\overline{I}$ is set to be $\overline{I} \cup \{i\}$. If $\overline{I} = \{0, 1, \ldots, k-1\}$, then $(c^*_0, c^*_1, \ldots, c^*_{k-1})\,\boldsymbol{G}$ is the minimum cost codeword and the algorithm terminate; otherwise, MA^* continues to search for an optimal path in the code tree until list $\boldsymbol{\mathcal{L}}$ is empty.

Every time the modified H^* determines the value of a component of $\boldsymbol{c}^*$, the code tree is halved. For an index $i \in I$, if the i^{th} component of $\boldsymbol{c}^*$ is determined and fixed by the modified H^*, then every path in the code tree whose i^{th} label is not equal to c_i^* can be discarded. It follows that all nodes in those paths at level greater than or equal to i can be discarded and no further search through these nodes is necessary.

6 Simulation results for the AWGN channel

We present simulation results for both the $(128, 64)$ and the $(256, 131)$ binary extended BCH codes when these codes are transmitted over AWGN channels. We assume that antipodal signaling is used in the transmission and we use the SNR per transmitted information bit γ_b in our simulation. For AWGN channels, $\boldsymbol{\phi} = \frac{4\sqrt{E}}{N_0}\,\boldsymbol{r}$ [9], so we can substitute $\boldsymbol{r}$ for $\boldsymbol{\phi}$ in our decoding algorithm.

A good initial upper bound on the cost of an optimal path will make our decoding algorithms more efficient. We generate the initial upper bound as follows. (1) We quantize the k-most reliable linearly independent components of the received vector and multiply this k-tuple by $\boldsymbol{G}$ to get a codeword. (2) Based on this codeword we generate all 2^b codewords by changing the values of the b least reliable components among the k-most reliable linearly independent components, and update the upper bound if we find a better upper bound. (3) We apply the order-3 decoding procedure introduced in [4] and update the upper bound if we find a better upper bound. Furthermore, we also incorporate the criterion introduced in [12] such that if this criterion is satisfied, the minimum cost codeword has been found and no further decoding is necessary.

We do not know HW for the $(128, 64)$ code, so we use a superset for it. We know that $d_{\min} = 22$ and that the Hamming weight of any codeword is divisible by 2 [10]. Thus, for this code, the superset used is $\{x| \ (x \text{ is even and } 22 \le x \le 106) \text{ or } (x = 0) \text{ or } (x = 128)\}$. The simulation results for the $(128, 64)$ code for $\gamma_b = 1.0$ dB, 1.5 dB, 2.0 dB, 2.5 dB, and 3.0 dB are given in Figure 1 (Hybrid) and Table 1 when q is set to be 4 and $M_B = 10000$. A lower bound on the bit error probability of the MLSD decoding algorithm is also given (LowerBound) [2]. The bit error probability for undecoded data is also given. "Quantized" data are obtained as follows. We quantize the k-most reliable linearly independent components of the received vector and get a k-tuple vector, multiply this k-tuple vector with the generator matrix $\boldsymbol{G}$, and get a codeword as the decoded codeword.

The simulation results for γ_b equal to 3.0 dB, 4.0 dB, 5.0 dB, 6.0 dB, 7.0 dB, and 8.0 dB are given in Table 2. These results were obtained by simulating 35,000 samples for each SNR when q is set to be 4 and M_B equal to 6000. No decoding error occurred during simulation.

From Figure 1, for the $(128, 64)$ code, the performance of our SOSD hybrid decoding algorithm is within 0.15 dB of the lower bound of the performance of the MLSD decoding algorithm. Thus, for the samples tried, limiting the size of list $\boldsymbol{\mathcal{L}}$ to 10000 nodes introduced only a small degradation on the performance of the algorithm for the $(128, 64)$ code.

In the tables, $\gamma_b = E_b/N_0$, N($\boldsymbol{r}$) = number of nodes visited during the decoding of $\boldsymbol{r}$, C($\boldsymbol{r}$) = number of codewords constructed during the decoding of $\boldsymbol{r}$, M($\boldsymbol{r}$) = number of nodes stored in list $\boldsymbol{\mathcal{L}}$ during the decoding of $\boldsymbol{r}$, max = maximum value among samples tried, ave = average value among samples tried, M_B is the upper bound on the size of list $\mathcal{L}$ in MA^*, p is the number of vectors generated by H^* during the decoding process, and b is the number of least reliable components considered among the k-most reliable linearly independent components in generating the initial upper bound.

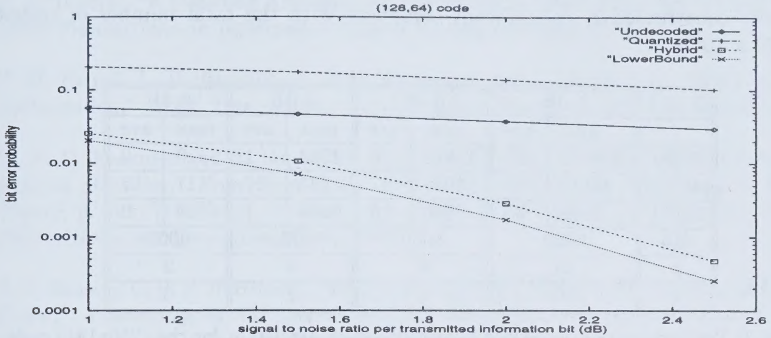


Figure 1: Performance of the SOSD hybrid decoding algorithm for the (128, 64) code

γ_b	1.0 dB		1.5 dB		2.0 dB		2.5 dB	
	max	ave	max	ave	max	ave	max	ave
N(r)	212534	110538	211202	91548	195386	66396	208931	39858
C(r)	180821	106939	196349	96156	161983	75914	159181	64549
M(r)	10000	8882	10000	7555	10000	5632	10000	3461
M_B	10000		10000		10000		10000	
p	256		256		128		128	
b	14		14		13		13	

Table 1: Performance of the SOSD hybrid decoding algorithm for the (128, 64) code

γ_b	3 dB		4 dB		5 dB		6 dB		7 dB		8 dB	
	max	ave	max	ave	max	ave	max	ave	max	ave	max	ave
N(r)	187410	18528	138196	1461	74685	34	1189	1	364	1	0	0
C(r)	141587	52718	102803	45285	27092	23	110	17	67	17	16	16
M(r)	60000	1638	10000	130	6000	6	882	1	315	1	0	0
M_B	60000		10000		6000		6000		6000		6000	
p	64		32		8		4		2		1	
b	12		11		10		4		2		0	

Table 2: Performance of the SOSD hybrid decoding algorithm for the (128, 64) code

We do not know HW for the (256, 131) code also, so we use a superset for it. We know that $d_{\min} = 38$ and that the Hamming weight of any codeword is divisible by 2. Thus for this code the superset used is $\{x| (x \text{ is divisible by } 2 \text{ and } 38 \leq x \leq 218), \text{ or } (x = 0), \text{ or } (x = 256)\}$. The simulation results for the $(256, 131)$ code for γ_b equal to 5.0 dB, 6.0 dB, 7.0 dB, and 8.0 dB are given in Table 3 when q is set to be 8 and M_B is equal to 6000. These results were obtained by simulating 35,000 samples for each SNR. No decoding error occurred during simulation. Furthermore, for the examples tried, the average number of codewords constructed is insignificant compared with the total number of codewords, $(2^{131} \simeq 2.72 \times 10^{39})$.

γ_b	5 dB		6 dB		7 dB		8 dB	
	max	ave	max	ave	max	ave	max	ave
N(r)	204432	227	14982	6	1784	1	0	0
C(r)	44874	41	1697	17	133	17	17	17
M(r)	6000	36	6000	5	6000	1	0	0
M_B	6000		6000		6000		6000	
p	16		8		4		2	
b	10		4		2		0	

Table 3: Performance of the SOSD hybrid decoding algorithm for the $(256, 131)$ code

7 Conclusion

In this paper we have proposed two novel decoding techniques and a hybrid version of these two techniques. MA^* is a graph-search algorithm which originates from A^* in Artificial Intelligence [8]. Simulation results have shown that MA^* is very efficient and has near-optimal performance for codes of short-to-moderate lengths citeSHIH95, but becomes impractical for long block codes transmitted over channels with low SNR since the memory requirement is large. Our new algorithm H^* is very flexible and can be incorporated with other techniques to reduce the search space during the decoding process. For example, simulation results for the (128,64) code transmitted over channels with $\gamma_b = 6$ dB, the average number of components of a minimum cost codeword determined by H^* for the samples tried during the decoding process is 61, and the search space is reduced from 2^{64} to $2^{64-61} = 2^3$. However, for long codes transmitted over channels with low SNR, the number of vectors needed to be generated in order to fix a component of the the minimum cost codeword grows exponentially.

The hybrid algorithm takes advantage of the strengths of these two algorithms and makes the decoding more efficient. The strength of H^* can overcome the weakness of MA^*, and vice versa. For codes of long lengths transmitted over low SNR, the memory requirement of MA^* is impractical while the memory requirement of H^* is small. The number of vectors needed to be generated before H^* can fix a component of the minimum cost codeword depends on the quality of the upper bound, and MA^* gives this upper bound during decoding process. Simulation results for the above linear block codes show that this hybrid decoding technique drastically reduces the search space, especially for most practical communication systems where the probability of error is less than 10^{-3} (γ_b greater than 6.8 dB) [1].

References

[1] G. C. Clark, Jr. and J. B. Cain, *Error-Correction Coding for Digital Communications.* New York, NY: Plenum Press, 1981.

[2] B. G. Dorsch, "A Decoding Algorithm for Binary Block Codes and J-ary Output Channels," *IEEE Transactions on Information Theory*, pp. 391–394, May 1974.

[3] K. H. Farrell, L. D. Rudolph, C. R.P. Hartmann, and L. D. Nielsen, "Decoding by Local Optimization," *IEEE Transactions on Information Theory*, pp. 740–743, September 1983.

[4] M. P. C. Fossorier and S. Lin, "Soft Decision Decoding of Linear Block Codes Based on Ordered Statistics," *Proceedings of the 1994 IEEE International Symposium on Information Theory*, p. 395, Trondheim, Norway, June 1994. Also submitted for publication in the *IEEE Transactions on Information Theory*, 1994.

[5] Y. S. Han and C. R. P. Hartmann, "Efficient Optimal and Suboptimal Decoding of Linear Block Codes," *Proceedings of the 1994 SBT/IEEE International Telecommunications Symposium (ITS '94)*, Rio de Janeiro, Brazil, August 1994.

[6] Y. S. Han, C. R. P. Hartmann, and C.-C. Chen, "Efficient Priority-First Search Maximum-Likelihood Soft-Decision Decoding of Linear Block Codes," *IEEE Transactions on Information Theory*, pp. 1514–1523, September 1993.

[7] Y. S. Han, C. R. P. Hartmann, and K. G. Mehrotra, "Further Results on Decoding Linear Block Codes Using a Generalized Dijkstra's Algorithm," *Proceedings of the 1994 IEEE International Symposium on Information Theory*, p. 342, Trondheim, Norway, June 1994.

[8] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transations on SSC*, vol. SSC-4, pp. 100–107, 1968.

[9] T.-Y. Hwang, "Decoding Linear Block Codes for Minimizing Word Error Rate," *IEEE Transactions on Information Theory*, pp. 733–737, November 1979.

[10] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes.* New York, NY: Elsevier Science Publishing Company, Inc., 1977.

[11] C.-C. Shih, C. R. Wulff, C. R. P. Hartmann, and C. K. Mohan, "Decoding Linear Block Codes Using Optimization Techniques," *Proceedings of the 1995 IEEE International Symposium on Information Theory*, p. 414, Whistler, British Columbia, Canada, September 1995.

[12] D. J. Taipale and M. B. Pursley, "An Improvement to Generalized-Minimum-Distance Decoding," *IEEE Transactions on Information Theory*, pp. 167–172, January 1991.

A Recursive Maximum Likelihood Decoding Procedure for a Linear Block Code Using an Optimum Sectionalized Trellis Diagram*

Toru Fujiwara† Hiroshi Yamamoto† Tadao Kasami†† Shu Lin†††

† Faculty of Engineering Science, Osaka University
Toyonaka, Osaka 560 Japan
†† Graduate School of Information Science, Nara Institute of Science and Technology
Ikoma, Nara 630-01 Japan
††† Department of Electrical Engineering, University of Hawaii at Manoa
Honolulu, Hawaii 96822 U.S.A.

ABSTRACT A new trellis-based MLD algorithm for a linear block code using a sectionalized trellis diagram is proposed and its computational complexity is analyzed in terms of the number of addition equivalent operations. In the proposed algorithm, to process the label sets of parallel branches in a section of the trellis, a maximum likelihood decoding procedure is used. The overall computational complexity depends on how to choose recursive sectionalization in the algorithm. A method for finding the optimum sectionalization of a trellis in terms of computation complexity using a dynamic programming approach is given. Numerical results show that a proper sectionalization of a trellis considerably reduces the decoding complexity.

1. Introduction

A trellis diagram for a linear block code is generally expressed and constructed in terms of sections [1]–[8]. A binary (N, K) block code C has an N-section trellis diagram in which each branch represents a code bit. An l-section trellis diagram with $l < N$ can be constructed from the N-section trellis diagram. In this l-section trellis diagram, each branch represents multiple code bits. In many cases, the trellis diagrams for linear block codes, such as Reed-Muller codes, are loosely connected and have simple parallel structure. The parallel structure of the l-section minimal trellis diagram for a binary linear block code was first investigated in [4], in which the number of parallel and structurally identical (except branch labels) sub-trellis diagrams without cross connections, called parallel components, in each section was expressed in terms of the dimensions of specific linear codes related to the given code.

It is shown in [5] that sectionalization of a trellis diagram does not reduce the branch complexity. However, if the section length is not too small, there are parallel branches between two adjacent states. These parallel branches may form a sub-trellis, and many adjacent state pairs have the same label set for the parallel branches. These structural properties can be used to reduce the computational complexity of a trellis-based maximum likelihood decoding (MLD) using the Viterbi algorithm. Furthermore, an MLD procedure can be used for finding the branches with the largest metrics among the sets of parallel branches in a section.

In this paper, a new trellis-based MLD algorithm for a linear block code using a sectionalized trellis diagram is proposed and its computational complexity is analyzed in terms of the number of addition-equivalent operations. The overall computational complexity depends on how to choose recursive sectionalization in the algorithm. A method for finding the optimum sectionalization of a trellis diagram in terms of computational complexity is given using a dynamic programming approach. Numerical results show that a proper sectionalization of a trellis diagram considerably reduces the decoding complexity.

---

*A preliminary version of this paper was presented at an IEICE meeting at Ikoma, Nara, Japan on July 22, 1995. The most part of this paper except for the comparison of the results with those in [9] has been submitted to IEICE Trans. Fundamentals [15].

In [9], an algorithm for optimizing the sectionalization of a trellis is given for Viterbi algorithm or variants of thereof. The decoding algorithm studied in [9] does not make use of the fact that many adjacent state pairs have the same label set for the parallel branches, and the decoding algorithm is not used recursively. The complexity of the proposed optimized decoding algorithm is not worse than that of the optimized one in [9] (refer to the end of Section 4). For example, the complexity for the third order Reed-Muller code of length 64 is reduced by 40%.

2. Structure of Minimal Trellis Diagram

For simplicity, we only consider binary linear block codes in this paper. The extension to nonbinary linear codes is straightforward.

Consider a binary linear (N, K) block code C. By an N-section trellis diagram for C [1]–[4], we mean a directed graph, denoted T, such that (1) T has an initial state s_0 and a final state s_f, (2) each branch has a label and two branches diverging from the same state have different labels, and (3) there is a directed path from s_0 to s_f with label sequence $u_1u_2\cdots u_N$ if and only if $(u_1, u_2, \ldots, u_N)$ is in C. A trellis diagram for C with the minimum number of states is said to be minimal, and a minimal trellis is unique within graph isomorphism [6].

For a nonnegative integer h not greater than N, let $S_h[C]$ denote the set of states of T just after the h-th bit position, where $S_0[C]$ consists of the initial state s_0 only and $S_N[C]$ consists of the final state s_f only. For a set of integers $U = \{h_0, h_1, h_2, \ldots, h_l\}$ with $0 = h_0 < h_1 < \cdots < h_l = N$, a sectionalized minimal trellis diagram for C, denoted $T_C(U)$ or simply $T(U)$, can be obtained from the N-section minimal trellis diagram T by deleting every state in $S_h[C]$ for $h \in \{0, 1, \ldots, N\} - U$ and every branch to or from a deleted state and by writing a branch with label α from a state $s \in S_{h_{j-1}}[C]$ to a state $s' \in S_{h_j}[C]$ for $1 \le j \le l$, if and only if there is a path with label α from s to s' in T [4].

Structural properties of a sectionalized trellis diagram of a linear block code have been analyzed in [4, 7]. Here, we briefly review them. Let h and h' be two integers such that $0 \le h < h' \le N$. For a binary N-tuple $\mathbf{v} = (v_1, v_2, \ldots, v_N)$, let $p_{h,h'}\mathbf{v}$ denote the binary $(h'-h)$-tuple $(v_{h+1}, v_{h+2}, \ldots, v_{h'})$ and let $p_{h,h'}[C]$ be defined as

$$p_{h,h'}[C] \triangleq \{p_{h,h'}\mathbf{v} : \mathbf{v} \in C\}. \tag{1}$$

Let $C_{h,h'}$ be the linear subcode of C consisting of all codewords whose components are all zero except for the $h'-h$ components from the $(h+1)$-th bit position to the h'-th bit position. Let $K_{h,h'}[C]$ be the dimension of $C_{h,h'}$. For convenience, $K_{h,h}[C]$ is defined as zero. For simplicity, we write $C^{\mathrm{tr}}_{h,h'}$ for $p_{h,h'}[C_{h,h'}]$, the truncation of $C_{h,h'}$. For integers h, h' and h'' such that $0 \le h < h' < h'' \le N$, let $K_{h,h',h''}[C]$ be defined as

$$K_{h,h',h''}[C] \triangleq K_{h,h''}[C] - K_{h,h'}[C] - K_{h',h''}[C]. \tag{2}$$

For simplicity, we write $K_h[C]$ for $K_{0,h,N}[C]$.

For a linear code A and its linear subcode B, let A/B denote the set of cosets in A with respect to B. For two states $s \in S_h[C]$ and $s' \in S_{h'}[C]$ in the minimal trellis diagram T of C, let $L(s, s')$ denote the set of all label sequences for paths from s to s'. Then,

$$\{L(s, s') : s \in S_h[C], s' \in S_{h'}[C]\} = p_{h,h'}[C]/C^{\mathrm{tr}}_{h,h'}, \tag{3}$$

and it has been shown in [3, Appendix A] that $|S_h[C]| = 2^{K_h[C]}$. For a state $s' \in S_{h'}[C]$, the number of those states s in $S_h[C]$ that $L(s, s') \neq \emptyset$ is given by $2^{Q_{h,h'}[C]}$ where

$$Q_{h,h'}[C] \triangleq \log_2 |p_{0,h}[C_{0,h'}]| - \log_2 |C^{\mathrm{tr}}_{0,h}| = K_{0,h,h'}[C]. \tag{4}$$

For a sectionalized trellis diagram $T(U)$ with $h, h' \in U$ and $h < h'$, consider the sub-trellis diagram obtained by truncating $T(U)$ except for the h-th to the h'-th bit positions. This truncated sub-trellis diagram, denoted $p_{h,h'}[T(U)]$, consists of parallel and structurally identical (except branch labels) sub-trellis diagrams without cross connections, called parallel components. The number of parallel components is given by $2^{K_h[C]-Q_{h,h'}[C]}$[4].

Let $C_{\overline{h,h'}}$ denote the linear subcode of C consisting of all codewords whose components from the $(h+1)$-th bit position to the h'-th bit position are all zero, and let $K_{\overline{h,h'}}[C]$ denote $\log_2 |C_{\overline{h,h'}}|$, the dimension of $C_{\overline{h,h'}}$. Then, the number of different label sequences in the truncated trellis diagram $p_{h,h'}[T]$ is given by [3, Appendix A]

$$|p_{h,h'}[C]| = 2^{K-K_{\overline{h,h'}}[C]}. \tag{5}$$

The number of different $L(s, s')$'s with $s \in S_h$ and $s' \in S_{h'}$ is given by $2^{K-K_{h,h'}[C]-K_{\overline{h,h'}}[C]}$. Any coset in $p_{h,h'}[C]/C^{\text{tr}}_{h,h'}$ appears $2^{K_h[C]+K_{\overline{h,h'}}[C]-K}$ times as the label set of parallel branches between state pairs in the section. For example, consider the 4-section minimal trellis diagram $T(\{0, 16, 32, 48, 64\})$ for the third order Reed-Muller code of length 64. In the second (or third) section, each coset in $p_{16,32}[C]/C^{\text{tr}}_{16,32}$ (or $p_{32,48}[C]/C^{\text{tr}}_{32,48}$) appears 64 times as the label set of parallel branches.

3. New Maximum Likelihood Decoding

3.1 Decoding Procedure

By using an l-section minimal trellis diagram $T(\{h_0, h_1, \ldots, h_l\})$ for C, a maximum likelihood decoding for C can be done as follows: Choose an integer m with $0 \le m \le l$. For every state in $S_{h_j}[C]$ with $1 \le j \le m$, find the survivor path from the initial state s_0 to the state with the largest metric together with its metric from the metrics of states in $S_{h_{j-1}}[C]$ in the following way: For every parallel branch set from a state in $S_{h_{j-1}}[C]$ to one in $S_{h_j}[C]$, find the branch with the largest metric. Then, find the survivor path by using the largest branch metrics of the parallel branches between every state pair in each parallel component. Similarly, for every state in $S_{h_j}[C]$ with $l-1 \ge j \ge m$, find the survivor path from the final state s_f to the state with the largest metric together with its metric from the metrics of states in $S_{h_{j+1}}[C]$. Then, for every state in $S_{h_m}[C]$, by concatenating survivor paths to the state, we obtain $|S_{h_m}[C]|$ paths from the initial state to the final state. Finally, find the path with the largest metric among these paths. This method is called the double-ended trellis decoding [10], and is denoted DETD($T(\{h_0, h_1, \ldots, h_l\}), h_m$). When m is chosen to l, the method corresponds to one-way (left to right) Viterbi decoding for $T(\{h_0, h_1, \ldots, h_l\})$.

In case that the same label set appears many times in a section as stated in Section 2, the decoding complexity can be reduced by constructing a table for the metrics of parallel branches, denoted PBT_{h_{j-1},h_j}, which, for each coset in $p_{h_{j-1},h_j}[C]/C^{\text{tr}}_{h_{j-1},h_j}$, stores the largest metric in the coset and a label with the metric.

Now, we devise the procedure to construct the table $\text{PBT}_{x,y}$ for integers x, y with $0 \le x < y \le N$. Since the set of cosets $p_{0,N}[C]/C_{0,N} = C/C$ consists of C only, the table $\text{PBT}_{0,N}$ stores the decoded codeword as well as its metric. Hereafter, the bit positions for $p_{x,y}[C]$ and its subcodes are numbered $x, x+1, x+2, \ldots, y$ instead of $0, 1, 2, \ldots, y-x$.
(A) A straightforward method to construct the table $\text{PBT}_{x,y}$ is to compute the metrics for all vectors in $p_{x,y}[C]$, and then find the vector with the largest metric for every coset by comparing the metrics of vectors in the coset. In general, this method is efficient only when $y - x$ is small.
(B) When $y - x$ is large, we can reduce the computational complexity by using a trellis diagram for each coset.

(B-a) Let l' be an integer with $2 \le l' \le y-x$ and $U' = \{h'_0, h'_1, \ldots, h'_{l'}\}$ be a set of integers with $x = h'_0 < h'_1 < \cdots < h'_{l'} = y$. Let $T_{C^{\rm tr}_{x,y}}(U')$ be an l'-section trellis diagram for $C^{\rm tr}_{x,y}$. For a coset $D = \mathbf{u} + C^{\rm tr}_{x,y}$ with $\mathbf{u} \in p_{x,y}[C]$, an l'-section minimal trellis diagram for D, denoted $T_D(U')$ can be obtained from $T_{C^{\rm tr}_{x,y}}(U')$ by adding $p_{h'_{j-1},h'_j}\mathbf{u}$ to the label of each branch from a state in $S_{h'_{j-1}}[C^{\rm tr}_{x,y}]$ to a state in $S_{h'_j}[C^{\rm tr}_{x,y}]$ for $1 \le j \le l'$. The set of states of $T_D(U')$ just after the h'_j-th bit position is also denoted $S_{h'_j}[D]$. The tables $\mathrm{PBT}_{h'_{j-1},h'_j}$ with $1 \le j \le l'$ are used for the decoding. These tables are constructed recursively.

(B-b) Instead of performing one-way (left to right) Viterbi decoding for each coset separately, we can execute the one-way Viterbi decoding with the following one trellis diagram for $p_{x,y}[C]/C^{\rm tr}_{x,y}$: In the trellis diagram, there is a common initial state $s_{x,y,0}$ and for each coset $D \in p_{x,y}[C]/C^{\rm tr}_{x,y}$, there is a final state $s_{x,y,f}[D]$ to which the set of label sequences of paths from $s_{x,y,0}$ is D. The trellis diagram can be constructed as follows: Choose a basis $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{K-K_{\overline{x,y}[C]}}$ of $p_{x,y}[C]$ such that the last $K_{x,y}[C]$ vectors are the basis of $C^{\rm tr}_{x,y}$. Let

$$\mathbf{u}_i \triangleq \begin{cases} \mathbf{v}_i \cdot \mathbf{e}_{i,K-K_{\overline{x,y}}[C]-K_{x,y}[C]}, & \text{for } 1 \le i \le K - K_{\overline{x,y}}[C] - K_{x,y}[C], \\ \mathbf{v}_i \cdot \mathbf{0}_{K-K_{\overline{x,y}}[C]-K_{x,y}[C]}, & \text{for } K - K_{\overline{x,y}}[C] - K_{x,y}[C] < i \le K - K_{\overline{x,y}}[C], \end{cases}$$

where $\mathbf{e}_{i,n}$ is the vector of length n such that only the i-th element is 1, $\mathbf{0}_n$ is the all-zero vector of length n, and '$\cdot$' denotes the concatenation of two vectors. Let $C_{\rm L}(x,y)$ be the binary linear code of length $n_{x,y} \triangleq y - x + K - K_{\overline{x,y}}[C] - K_{x,y}[C]$ spanned by $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{K-K_{\overline{x,y}}[C]}$. The bit positions for $C_{\rm L}(x,y)$ are numbered $x, x+1, \ldots, y, \ldots, x+n_{x,y}$ instead of $0, 1, \ldots, y-x, \ldots, n_{x,y}$. Then, the sub-trellis diagram obtained by truncating an $(l'+1)$-section minimal trellis diagram of $C_{\rm L}(x,y)$, $T_{C_{\rm L}(x,y)}(U' \cup \{x + n_{x,y}\})$, except for the x-th to the y-th bit positions is the desired l'-section trellis diagram. This truncated trellis diagram is denoted $p_{x,y}[T_{C_{\rm L}(x,y)}(U')]$ for simplicity.

Define $z = h'_{l'-1}$. Consider the one-way (left to right) Viterbi decoding using the trellis diagram $T_{C_{\rm L}(x,y)}(\{x,z,y\})$ together with the two tables $\mathrm{PBT}_{x,y}$ and $\mathrm{PBT}_{z,y}$. We compare the decoding complexity of this method with that using $p_{x,y}[T_{C_{\rm L}(x,y)}(U')]$. It follows from (3), the definition of $C_{\rm L}(x,y)$ that for any u and z with $x \le u < z \le y$,

$$\{L(s,s') : s \in S_u[C_{\rm L}(x,y)]\}, s' \in S_z[C_{\rm L}(x,y)]\} = p_{u,z}[C]/C^{\rm tr}_{u,z}. \tag{6}$$

From this fact and that the minimality of $p_{x,y}[T_{C_{\rm L}(x,y)}(U')]$, we have that $p_{x,z}[p_{x,y}[T_{C_{\rm L}(x,y)}(U')]] = p_{x,z}[T_{C_{\rm L}(x,y)}(U')]$ is the $(l'-1)$-section minimal trellis diagram for $p_{x,z}[C]/C^{\rm tr}_{x,z}$, $p_{x,z}[T_{C_{\rm L}(x,z)}(\{h'_0, h'_1, \ldots, z\})]$. Therefore, the complexity of the decoding with $p_{x,y}[T_{C_{\rm L}(x,y)}(\{x,z,y\})]$ is not worse than that with $p_{x,y}[T_{C_{\rm L}(x,y)}(U')]$, and it is enough to consider the two-section trellis diagram $T_{C_{\rm L}(x,y)}(\{x,z,y\})$ only at each recursion level, for finding a decoding procedure with the smallest complexity, as far as the complexity depends on the used trellis diagram only.

(B-c) The above method using the two-section trellis diagram $T_{C_{\rm L}(x,y)}(\{x,z,y\})$ can be viewed as an algorithm to solve the following problem: how to obtain $\mathrm{PBT}_{x,y}$ from $\mathrm{PBT}_{x,z}$ and $\mathrm{PBT}_{z,y}$ efficiently, where $x < z < y$. Since $|C^{\rm tr}_{x,y}/(C^{\rm tr}_{x,z} + C^{\rm tr}_{z,y})| = 2^{K_{x,z,y}[C]}$, for $D \in p_{x,y}[C]/C^{\rm tr}_{x,y}$, we can find the set of $2^{K_{x,z,y}[C]}$ coset pairs in $p_{x,z}[C]/C^{\rm tr}_{x,z} \times p_{z,y}[C]/C^{\rm tr}_{z,y}$, denoted $\mathrm{CP}[D]$, such that

$$D = \bigcup_{(D_{\rm L},D_{\rm R}) \in \mathrm{CP}[D]} D_{\rm L} \cdot D_{\rm R} \triangleq \bigcup_{(D_{\rm L},D_{\rm R}) \in \mathrm{CP}[D]} \{\mathbf{u} \cdot \mathbf{v} : \mathbf{u} \in D_{\rm L}, \mathbf{v} \in D_{\rm R}\}. \tag{7}$$

For a coset X, let $m[X]$ denote the largest metrics of X. From (7),

$$m[D] = \max_{(D_{\rm L},D_{\rm R}) \in \mathrm{CP}[D]} \{m[D_{\rm L}] + m[D_{\rm R}]\}. \tag{8}$$

Let $N(x, y; z)$ be the number of addition-equivalent operations to find $m[D]$, when the tables $\mathrm{PBT}_{x,z}$ and $\mathrm{PBT}_{z,y}$ are given. $N(x, y; z)$ depends on the used algorithm. If we simply follow the right-hand side of (8), the value of $N(x, y; z)$, denoted $N^{(0)}(x, y; z)$, is given by

$$N^{(0)}(x, y; z) = 2^{K-K_{\overline{x,y}}[C]-K_{x,y}[C]}(2^{K_{x,z,y}[C]+1} - 1). \tag{9}$$

We review the method using the two-section trellis diagram $T_{C_{\mathrm{L}}(x,y)}(\{x, z, y\})$. Let s' be the state in $S_y[C_{\mathrm{L}}(x, y)]$ corresponding to D, that is, $L(s_{x,y,0}, s') = D$. Then, from (6), $\mathrm{CP}[D]$ is given by $\{((L(s_{x,y,0}, s), L(s, s')) : s \in S_z[C_{\mathrm{L}}(x, y)]$ and $L(s, s') \neq \emptyset\}$. Therefore, a standard processing of the trellis diagram $T_{C_{\mathrm{L}}(x,y)}(\{x, z, y\})$ to find $m[D]$ is essentially the same as the method which simply follows the right-hand side of (8).

We can also use a trellis diagram for $p_{x,y}[C]/C^{\mathrm{tr}}_{x,y}$ such that there is a common final state $s_{x,y,f}$ and for each coset $D \in p_{x,y}[C]/C^{\mathrm{tr}}_{x,y}$, there is a initial state $s_{x,y,0}[D]$. We can also show that this method is also the essentially same as the above method.

There is another implementation [7, 12] based on the detailed structure of each section that is a union of the same simple regular sub-trellis diagrams. The implementation yields a smaller $N(x, y; z)$ for the worst case than $N^{(0)}(x, y; z)$ and its average computational complexity is considerably smaller than the worst case one.

By summarizing the above, we devise the following procedure $NewMLD(x, y)$ to construct the table $\mathrm{PBT}_{x,y}$ for a received word.

[Procedure $NewMLD(x, y)$**]**

Input: Integers x, y. Output: $\mathrm{PBT}_{x,y}$.

(Step 1) Choose an integer z. If $x + 1 = y$, let $z = -1$. Otherwise, choose an integer z with $z = -1$ or $x < z < y$.

(Step 2) Execute one of the following (Case A) or (Case B) depending on the value of z. When $z = -1$, execute (Case A). Otherwise, execute (Case B). In both cases, we obtain the table $\mathrm{PBT}_{x,y}$ by storing the results.

(Case A: $z = -1$) Compute the metrics for all vectors in $p_{x,y}[C]$. The method of this metric computation is discussed in 3.2. For every coset in $p_{x,y}[C]/C^{\mathrm{tr}}_{x,y}$, find the vector with the largest metric by comparing the metrics of vectors in the coset.

(Case B: $x < z < y$)

(Step B-1) Construct the tables $\mathrm{PBT}_{x,z}$ and $\mathrm{PBT}_{z,y}$ by executing $NewMLD(x, z)$ and $NewMLD(z, y)$, respectively.

(Step B-2) For every coset $D \in p_{x,y}[C]/C^{\mathrm{tr}}_{x,y}$, find the vector with the largest metric in the coset by using (8). ΔΔ

3.2 Complexity Analysis and Optimum Sectionalization

In the following, we evaluate the computational complexity, denoted $\psi(x, y)$, of the above procedure $NewMLD(x, y)$ in terms of the number of addition-equivalent operations. This number includes those for all the recursive executions of $NewMLD$. The chosen integer z in (Step 1) for $NewMLD(x, y)$ is denoted by $z_{x,y}$. When it is necessary to specify the value of parameter $z_{x,y}$, notation $\psi(x, y; z_{x,y})$ is used.

(A) First, we derive a formula for $\psi(x, y; -1)$. We assume that the metrics of the individual bits are given.

1) Consider the case where $x + 1 = y$. Assume for the simplicity that the minimum distance of C is greater than 1. Then, no operation is needed and hence

$$\psi(x, x + 1; -1) = 0. \tag{10}$$

2) When $x + 1 < y$, the number $\psi(x, y; -1)$ depends on the implementation.

2.0) For example, if the branch metrics for all branch labels are computed independently, the value $\psi(x, y; -1)$ for this implementation, denoted $\psi^{(0)}(x, y; -1)$, is given by

$$\psi^{(0)}(x, y; -1) = (y - x - 1)2^{K-K_{\overline{x,y}}[C]} + 2^{K-K_{\overline{x,y}}[C]-K_{x,y}[C]}(2^{K_{x,y}[C]} - 1). \tag{11}$$

2.1) We may also use the following method to compute the branch metrics for labels in $p_{x,y}[C]$. Consider the case $|p_{x,y}[C]| = 2^{y-x}$. First, compute the metric of a label with $y-x-1$ additions. Then, we compute the metric of each branch label by one addition operation in the order of the Gray code [9, 11]. This method can also be used for the case where $|p_{x,y}[C]| < 2^{y-x}$. Let $\psi^{(1)}(x, y; -1)$ be the value of $\psi(x, y; -1)$ for this implementation. Then, we have that

$$\psi^{(1)}(x, y; -1) = 2^{y-x} + y - x - 2 + 2^{K-K_{\overline{x,y}}[C]-K_{x,y}[C]}(2^{K_{x,y}[C]} - 1). \tag{12}$$

2.2) Suppose that the all-one vector of length $y-x$, denoted $\mathbf{1}_{y-x}$, is always in $p_{x,y}[C]$ for any x, y with $0 \leq x < y \leq N$. This holds whenever C is a cyclic code whose generator polynomial has no factor $(X-1)$, the extended code of such a cyclic code, or its permuted code. All the example codes considered in Section 4 satisfy this condition. We use a technique in [9] to reduce the number of operations, where negation is assumed costless. The metrics of 2^{y-x-1} labels are computed first by using the above method in 2.2), and then the remaining metrics are computed by negating the first 2^{y-x-1} metrics. If $\mathbf{1}_{y-x} \in C^{\mathrm{tr}}_{x,y}$, we can reduce the number of comparisons by using the fact that the metric of a label in a coset of $C^{\mathrm{tr}}_{x,y}$ is the negation of the metric of a complementary label in the coset [9]. Note that this can also be used to reduce the number of comparisons in 2.1). Let $\psi^{(2)}(x, y; -1)$ be the value of $\psi(x, y; -1)$ for this implementation. Then, we have that

$$\begin{aligned}&\psi^{(2)}(x, y; -1)\\ &= \begin{cases} 2^{y-x-1} + y - x - 2 + 2^{K-K_{\overline{x,y}}[C]-K_{x,y}[C]}(2^{K_{x,y}[C]-1} - 1), & \text{if } \mathbf{1}_{y-x} \in C^{\mathrm{tr}}_{x,y}, \\ 2^{y-x-1} + y - x - 2 + 2^{K-K_{\overline{x,y}}[C]-K_{x,y}[C]}(2^{K_{x,y}[C]} - 1), & \text{otherwise.} \end{cases}\end{aligned} \tag{13}$$

Note that there is another method in [9] which is applicable when the dual code of $p_{x,y}[C]$ contains $\mathbf{1}_{y,x}$ and $y-x$ is even. In our decoding algorithm, we can show that this method does not reduce the decoding complexity for the optimum sectionalization.

(B) Next, consider the case with $z_{x,y} \neq -1$. We have the following recursive formula:

$$\psi(x, y; z_{x,y}) = \psi(x, z_{x,y}) + \psi(z_{x,y}, y) + N(x, y; z_{x,y}), \qquad \text{for } x < z_{x,y} < y. \tag{14}$$

We need choose z in (Step 1) properly to obtain a decoder with the small number of operations. A sectionalization of the trellis diagrams (choices of z) which gives the smallest $\psi(0, N)$ for a given code is called the optimum sectionalization for the code. Define

$$\begin{aligned}&\psi_{\min}(x, y) =\\ &\begin{cases} \psi(x, y; -1) = 0, & \text{if } x + 1 = y, \\ \min\{\psi(x, y; -1), \min\limits_{x<z<y}\{\psi_{\min}(x, z) + \psi_{\min}(z, y) + N(x, y; z)\}\}, & \text{otherwise.} \end{cases}\end{aligned} \tag{15}$$

$\psi_{\min}(x, y)$ is the smallest number of operations to construct the table $\mathrm{PBT}_{x,y}$, and therefore $\psi_{\min}(0, N)$ is that for decoding a received word. By using (15) together with formulas for $\psi(x, y; -1)$ and $N(x, y; z)$, we can compute $\psi_{\min}(x, y)$ for every (x, y) with $0 \leq x < y \leq N$, efficiently. By storing the information when the minimum value occurs in the right-hand side of (15) for $x + 1 < y$, it is easy to find an optimum sectionalization.

Finally, we present two theorems on the choice of (Case A) or (Case B). The proofs of them can be found in [15].

Theorem 1: Consider a binary linear code C of length N such that a generator matrix of C does not contain any all-zero column. Suppose that $\psi^{(0)}(x, y; -1)$ is used for $\psi(x, y; -1)$ and $N^{(0)}(x, y; z)$ is used for $N(x, y; z)$.

(1) The complexity for (Case B) is smaller than that for (Case A) except for $y - x \leq 2$. When $y - x = 2$, they are the same, if the minimum distance of C is greater than 1.

(2) If $y - x > 2$, $|p_{x,y}[C]| = 2^{y-x}$ and $C^{\text{tr}}_{x,y} = \{\mathbf{0}_{y-x}\}$ or $\{\mathbf{0}_{y-x}, (1, *, \ldots, *, 1)\}$, where $(1, *, \ldots, *, 1)$ denotes a vector $\mathbf{v}$ such that $p_{x,x+1}\mathbf{v} = p_{y-1,y}\mathbf{v} = 1$, then the right-hand side of (15) takes its minimum for both $z = \lfloor (x+y)/2 \rfloor$ and $z = \lceil (x+y)/2 \rceil$.

Theorem 2: Suppose that $\psi^{(1)}(x, y; -1)$ is used for $\psi(x, y; -1)$ and $N^{(0)}(x, y; z)$ is used for $N(x, y; z)$.

(1) If $|p_{x,y}[C]| = 2^{y-x}$ and $C^{\text{tr}}_{x,y} = \{\mathbf{0}_{y-x}\}$ or $\{\mathbf{0}_{y-x}, (1, *, \ldots, *, 1)\}$, then the complexity for (Case A) is smaller than that for (Case B) except for $y - x \leq 2$. When $y - x = 2$, they are the same.

(2) If the condition in (1) does not hold, then the complexity for (Case B) is smaller than that for (Case A), and if $|p_{x,y}[C]| < 2^{y-x}$ and $C^{\text{tr}}_{x,y} = \{\mathbf{0}_{y-x}\}$ or $\{\mathbf{0}_{y-x}, (1, *, \ldots, *, 1)\}$, then the right-hand side of (15) takes its minimum for both $z = \lfloor (x+y)/2 \rfloor$ and $z = \lceil (x+y)/2 \rceil$.

4. Examples

Let RM(64, 22), RM(64, 42) and RM(64, 57) be the second, third and fourth order Reed-Muller code of length 64, respectively. Let exBCH(63, k) denote the extended code of the binary primitive (63, k) BCH code. We computed the decoding complexity of RM(64, 22), RM(64, 42), RM(64, 57) and permuted codes of exBCH(63, k) with $10 \leq k \leq 51$.

A cyclic (or shortened cyclic) code or its extended code has the worst trellis state complexity among linear block codes of the same length and dimension [8]. In order to obtain a trellis diagram with a smaller number of states for these codes, the order of symbol positions must be permuted. The natural bit ordering of a Reed-Muller code is optimal for the state complexity [13]. For extended BCH codes, we consider the following permutations [8, 14]: Let α be a primitive element of GF(2^6) and let $\{\beta_1, \ldots, \beta_6\}$ be a basis of GF(2^6) over GF(2). For a positive integer i less than 2^6, let α^{i-1} be expressed as $\alpha^{i-1} = \sum_{j=1}^{6} b_{i,j}\beta_j$, with $b_{i,j} \in$ GF(2). For $i = 0$, let $b_{0,j} \triangleq 0$ for $1 \leq j \leq 6$. Let π denote the following permutations on $\{1, 2, \ldots, 2^6\}$, $\pi(i) \triangleq 1 + \sum_{j=1}^{6} b_{i-1,j}2^{6-j}$ for $1 \leq i \leq 2^6$. In general, the state complexity of the minimal trellis diagram for a permuted code depends on the choice of the basis $\{\beta_1, \beta_2, \ldots, \beta_6\}$. We consider the following three bases for codes of length 64: (1) Basis A is a polynomial basis, $\{1, \alpha, \alpha^2, \ldots, \alpha^5\}$. (2) Basis B is $\{1, \alpha, \alpha^2, \alpha^{21}, \alpha^{22}, \alpha^{23}\}$, which is obtained by combining a basis GF(2^6) over GF(2^2), $\{1, \alpha, \alpha^2\}$. and a basis GF(2^2) over GF(2), $\{1, \alpha^{21}\}$. (3) Basis C is $\{1, \alpha, \alpha^9, \alpha^{10}, \alpha^{18}, \alpha^{19}\}$, which is obtained by combining a basis of GF(2^6) over GF(2^3), $\{1, \alpha\}$, and a basis of GF(2^3) over GF(2), $\{1, \alpha^9, \alpha^{18}\}$.

In Table 1, the number of addition-equivalent operations are listed for the above example codes. We use $N^{(0)}(x, y; z)$ for $N(x, y, z)$. The column labeled 64-section gives the numbers of operations required in conventional Viterbi decoding with 64-section trellis diagram, which is equivalent to the proposed method with the sectionalization $z_{0,i} = i - 1$, with $2 \leq i \leq 64$. The value is given by $2|B| - |S| - 1$, where $|S|$ and $|B|$ is the total number of branches and states, respectively, in the entire 64-section trellis diagram. The column labeled Optimum ($\psi^{(i)}_{\min}$) with $0 \leq i \leq 2$ gives the number of operations for the optimum sectionalization when $\psi^{(i)}(x, y; -1)$ is used for $\psi(x, y; -1)$. These three optimum values for each code are almost the same. For every BCH code, the two bit permutations other than one indicated in Table 1 among the three bit permutations above-mentioned, the optimum values are larger than those in the table. It is worth while to mention that the numbers of operations required in conventional Viterbi decoding with 64-section trellis diagram for exBCH(63,51) takes its minimum value 340, 217 for Basis B among the three

Table 1: The number of addition-equivalent operations for the maximum likelihood decoding.

Code	64-section	Optimum ($\psi^{(0)}_{\min}$)	Optimum ($\psi^{(1)}_{\min}$)	Optimum ($\psi^{(2)}_{\min}$)	Lafourcade-Vardy[9]
RM(64,22)	425,209	78,209	78,119	77,896	101,786
RM(64,42)	773,881	326,017	325,717	323,759	538,799
RM(64,57)	7,529	5,281	5,209	4,999	6,507
exBCH(63,10), Basis C	20,073	3,201	3,201	3,108	4,074
exBCH(63,16), Basis B	764,153	120,193	120,103	119,880	148,566
exBCH(63,18), Basis B	2,865,401	468,353	468,263	468,040	509,120
exBCH(63,24), Basis B	1,327,353	271,745	271,655	271,432	316,608
exBCH(63,30), Basis C	35,028,985	16,091,009	16,090,739	16,056,668	16,598,063
exBCH(63,36), Basis C	18,710,521	9,995,617	9,995,419	9,961,580	12,829,263
exBCH(63,39), Basis C	38,436,857	24,741,161	24,740,975	24,707,149	30,982,731
exBCH(63,45), Basis C	1,082,105	893,489	893,237	891,695	891,819
exBCH(63,51), Basis A	418,553	312,721	312,673	312,382	393,528

permutations. This shows that a good bit ordering for the N-section trellis diagram is not always good for our decoding procedure. The last column shows the numbers of addition-equivalent operations given by Lafourcade and Vardy [9]. Note that the bit permutations in [9] for some BCH codes are different from those in this paper, although the bit ordering for RM codes are the same. Only for the last three BCH codes in Table 1, the values in the column labeled 64-section are the same as those in [9].

Optimum sectionalizations of several example codes when $\psi^{(2)}(x, y; -1)$ is used for $\psi(x, y; -1)$ are presented in Table 2. The optimum sectionalization is chosen in the following manner: If $\psi_{\min}(x, y) = \psi(x, y; -1)$, then $z_{x,y} = -1$. Otherwise, $z_{x,y}$ is the largest integer z such that $\psi_{\min}(x, y) = \psi_{\min}(x, z) + \psi_{\min}(z, y) + N(x, y; z)$. In the table, we present the values of $z_{x,y}$ for every (x, y) such that $NewMLD(x, y)$ is invoked during the execution of $NewMLD$ except for (x, y) with $z_{x,y} = -1$. The sectionalization (the structure of recursive executions of $NewMLD$ during the execution of $NewMLD(0, N)$) can be expressed as the following ordered binary tree R: A node of R represents each invocation of $NewMLD$. The node corresponds to the invocation $NewMLD(x, y)$ is labeled (x, y). The label of the root of R is $(0, N)$. The node labeled (x, y) has the left child labeled $(x, z_{x,y})$ and the right child labeled $(z_{x,y}, y)$, if and only if $z_{x,y} \neq -1$. This represents that $NewMLD(x, y)$ calls $NewMLD(x, z_{x,y})$ and $NewMLD(z_{x,y}, y)$. A node (x, y) in R is a leaf if $z_{x,y} = -1$. The sectionalization R can be reconstructed from the values of $z_{x,y}$ easily. For RM(64, 42), the tree R is depicted in Figure 1.

In the procedure $NewMLD$, we have considered only two-section trellis diagrams. As discussed in (B) of Section 3, the following reformulation of $NewMLD$ is essentially identical to the original procedure. In this reformulation, l-section trellis diagrams with $l \geq 2$ are used.
(Step 1') Choose an integer l with $l \geq 1$ and a set of $l + 1$ integers $U = \{h_0, h_1, \ldots, h_l\}$ with $x = h_0 < h_1 < \cdots < h_l = y$.
(Step 2') When $U = \{x, y\}$, execute (Case A) in the original $NewMLD$. Otherwise, execute the following (Step B-1') and (Step B-2').
(Step B-1') Construct the tables $\mathrm{PBT}_{h_{i-1},h_i}$ by executing the variation of $NewMLD$ recursively for $1 \leq i \leq l$.
(Step B-2') Choose an integer z in U. For every coset $D \in p_{x,y}[C]/C^{\mathrm{tr}}_{x,y}$, find the vector with the largest metric in the coset by $\mathrm{DETD}(T_{C_{\mathrm{L}}(x,y)}(U), z)$. $\Delta\Delta$

Table 2: Optimum sectionalization of the several example codes when $\psi^{(2)}(x, y; -1)$ is used for $\psi(x, y; -1)$.

Code	An Optimum Sectionalization
RM(64,22)	$z_{0,64} = 63, z_{0,63} = 61, z_{0,61} = 56, z_{0,56} = 48, z_{0,48} = 32, z_{32,48} = 40,$ $z_{0,32} = 16, z_{16,32} = 24, z_{0,16} = 8$
RM(64,42)	$z_{0,64} = 56, z_{0,56} = 48, z_{0,48} = 32, z_{32,48} = 40, z_{0,32} = 16, z_{16,32} = 24, z_{0,16} = 8$
exBCH(63,24) Basis B	$z_{0,64} = 63, z_{0,63} = 61, z_{0,61} = 56, z_{0,56} = 48, z_{0,48} = 32, z_{32,48} = 40,$ $z_{0,32} = 16, z_{16,32} = 24, z_{0,16} = 8$
exBCH(63,30) Basis C	$z_{0,64} = 63, z_{0,63} = 60, z_{0,60} = 32, z_{32,60} = 56, z_{32,56} = 48, z_{0,32} = 28,$ $z_{0,28} = 24, z_{0,24} = 16$
exBCH(63,45) Basis C	$z_{0,64} = 63, z_{0,63} = 62, z_{0,62} = 32, z_{32,62} = 61, z_{32,61} = 60, z_{32,60} = 59,$ $z_{32,59} = 58, z_{32,58} = 56, z_{32,56} = 48, z_{32,48} = 40, z_{0,32} = 30, z_{0,30} = 29,$ $z_{0,29} = 28, z_{0,28} = 27, z_{0,27} = 26, z_{0,26} = 24, z_{0,24} = 16, z_{0,16} = 8$

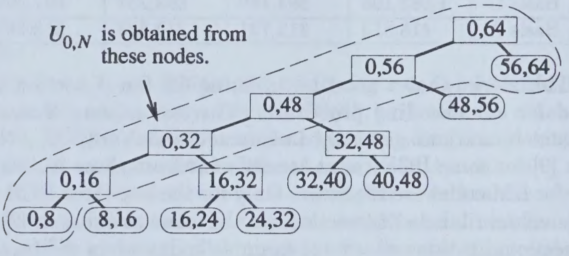


Figure 1: An optimum sectionalization for RM(64,42) when $\psi^{(2)}(x, y; -1)$ is used for $\psi(x, y; -1)$.

For a given sectionalization (binary tree) R for the original $NewMLD$, we can choose the parameters of the reformulated procedure as follows:

(1) For $(x, y) = (0, N)$, $U \triangleq \{0, N\} \cup \{i|$ a node labeled $(0, i)$ or (i, N) is in $R\}$ (refer to Fig. 1), and $z = z_{0,N}$.

(2) For each execution $NewMLD(h_{i-1}, h_i)$ in (Step B-1'), we choose $\{h_{i-1}, h_i\} \cup \{j|$ a node labeled (h_{i-1}, j) or (j, h_i) is in $R\}$ as U and $z_{x,y}$ as z. $\Delta\Delta$

Figure 2 illustrates the sectionalization of this reformulation for RM(64, 42).

The decoding algorithm considered in [9] is derived from the above reformulation of $NewMLD$ by (M1) restricting the choice of the parameter U in such a way that $U = \{x, y\}$ for any $(x, y) \neq (0, N)$, and (M2) modifying (Case A) as follows: find the vector with the largest metric for every $L(s, s')$ with $s \in S_x[C]$ and $s' \in S_y[C]$, instead of finding them only for the representatives label sets, $p_{x,y}[C]/C^{\rm tr}_{x,y}$. Most of the improvement on the complexity in the new procedure is due to the recursive execution of the procedure. The above modification of (Case A) contributes to the improvement little. For each example code, the optimum complexity for $\psi^{(0)}$ is not increased at all by modifying (Case A) only, and those for $\psi^{(1)}$ and $\psi^{(2)}$ are increased by less than 4.3%.

5. Conclusion

In this paper, a new efficient trellis-based maximum likelihood decoding algorithm for a linear block code using a sectionalized trellis diagram is proposed and its computational complexity is analyzed in terms of the number of addition equivalent operations. The

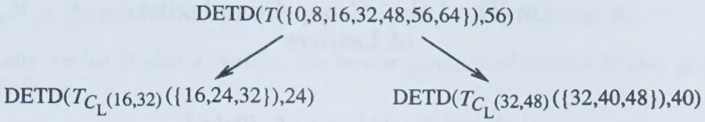

DETD(T({0,8,16,32,48,56,64}),56)
DETD(T_{C_L}(16,32)({16,24,32}),24)
DETD(T_{C_L}(32,48)({32,40,48}),40)

Figure 2: An implementation by using DETD for RM(64,42).

optimum sectionalizations of trellis diagram for some well known codes are found.

Acknowledgment

The authors would like to thank Professor Alexander Vardy for sending us their paper [9].

References

[1] L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decodomg of Linear Codes for Minimizing Symbol Error Rate," IEEE Trans. Inf. Theory, vol.IT-20, no.2, pp.284–287, Mar. 1974.

[2] J. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis," IEEE Trans. Inf. Theory, vol.IT-24, no.1, pp.76–80, Jan. 1978.

[3] G.D. Forney Jr., "Coset codes—part II: binary lattices and related codes," IEEE Trans. Inf. Theory, vol.34, no.5, pp.1152–1187, Sep. 1988.

[4] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "On structural complexity of the l-section minimal trellis diagrams for binary linear block codes," IEICE Trans. Fundamentals, vol.E76-A, no.9, pp.1411–1421, Sep. 1993.

[5] G.D. Forney Jr., "Dimension/length profiles and trellis complexity of linear block codes," IEEE Trans. Inf. Theory, vol.40, no.6, pp.1741–1752, Nov. 1994.

[6] D.J. Muder, "Minimal trellises for block codes," IEEE Trans. Inf. Theory, vol.34, no.5, pp.1049–1053, Sep. 1988.

[7] T. Kasami, T. Fujiwara, Y. Desaki, and S. Lin, "On branch labels of parallel components of the L-section minimal trellis diagrams for binary linear block codes," IEICE Trans. Fundamentals, vol.E77-A, no.6, pp.1058–1068, Jun. 1994.

[8] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "On complexity of trellis structure of linear block codes," IEEE Trans. Inf. Theory, vol.39, no.3, pp.1057–1064, May 1993.

[9] A. Lafourcade and A. Vardy, "Optimum Sectionalization of a Trellis," submitted to IEEE Trans. Inf. theory, 1995.

[10] B. Honary, G. Markarian, and S. Marple, "Trellis decoding of the Reed-Solomon codes: practical approach," Int'l Sympo. Commun. Theory & Applications, pp.12–16, Jul. 1995.

[11] A. Vardy and Y. Be'ery, "Maximum-likelihood soft decoding of BCH codes," IEEE Trans. Inf. theory, vol.40, no.2, pp.546–554, Mar. 1994.

[12] H. Nagano, T. Fujiwara, and T. Kasami, "Average complexity evaluation of an MLD algorithm using the trellis structure for a linear block code," IEICE Trans. Fundamentals, vol.E78-A, no.9, Sep. 1995.

[13] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "On the optimum bit orders with respect to the state complexity of trellis diagrams for binary linear codes," IEEE Trans. Inf. Theory, vol.39, no.1, pp.242–245, Jan. 1993.

[14] T. Fujiwara, T. Kasami, R. Morelos-Zaragoza, and S. Lin, "The state complexity of trellis diagram for a class of generalized concatenated codes," Proc. 16th Sympo. Inf. Theory and Its Applications, pp.21–24, Kanazawa, Japan, Oct. 1993.

[15] H. Yamamoto, T. Fujiwara, T. Kasami, and S. Lin, "A Recursive Maximum Likelihood Decoding Algorithm for a Linear Block Code Using a Sectionalized Trellis Diagram and Its Optimization," submitted to IEICE Trans. Fundamentals, 1995.

On The Label Group Complexity of Lattices

Vahid Tarokh[1] and I.F. Blake [2]

Abstract

We study the best trade-off between the coding gain and the label group complexity of rational lattices. Using Kitaoka's theorem, we establish that the numbers of the rounding operations as well as the squaring operations needed for decoding rational lattices grow at most linearly in terms of the coding gain. Thus, the exponential nature of the Viterbi decoding algorithm (when applied to the trellis diagram of lattices) originates from the comparison and addition operations.

The issues of trellis complexity recently has attracted wide attention. In this presentation, we consider the size of the label groups of the trellis diagrams of lattices. As established in [6], this relates to the number of rounding operations as well as the number of squaring operations required for the Viterbi-decoding of a given lattice. The results are a summary of those established in section 3.6 of [6].

Let $\mathcal{L}$ denote the category of all the lattices having a finite trellis diagram. Let $L \in \mathcal{L}$ be of dimension n and coding gain $\delta(L)$. Let $C(L)$ denote the category of all the trellis diagrams of L. Consider the sum of the cardinalities of label groups for each element of $C(L)$. Let G denote the minimum of these sums in $C(L)$. The label group complexity of L is defined by $\mathcal{G}(L) = G/n$ [6]. We consider the best trade off between the label group complexity and coding gain of elements of $\mathcal{L}$ by defining the label trellis complexity function as below:

$$\mathcal{T}_3(\gamma) \;=\; \inf\{\mathcal{G}(L) \mid \delta(L) \geq \gamma \text{ and } L \in \mathcal{L}\}. \tag{1}$$

We studied the behaviour of $\mathcal{T}_3$ in [6] and established that $\mathcal{T}_3$ is a continuous function with left and right derivatives existing everywhere. Moreover, $\mathcal{T}_3$ is differntiable almost everywhere with second derivative existing everywhere except on a set of measure zero. $\mathcal{T}_3$ is integral in the sense that

$$\int_a^b \mathcal{T}_3'(x)\, dx = \mathcal{T}_3(b) - \mathcal{T}_3(a),$$

whenever $1 \leq a \leq b < \infty$ [6].

In [6], an explicit closed form for $\mathcal{T}_3(\cdot)$, for small values of $\gamma \geq 1$ was derived. There also a tight lower bound on $\mathcal{T}_3$ was established. Here, we review the derivation of an upper bound on $\mathcal{T}_3$ given in [6]. The derivation starts with considering the tensor product of lattices.

Definition 1 *Let M and N be matrices and assume that $M = (m_{ij})$, $1 \leq i \leq r$, $1 \leq j \leq s$. The tensor product of M and N, denoted by $M \otimes N$, is defined to be*

$$\begin{pmatrix} m_{11}N & m_{12}N & \ldots & m_{1s}N \\ \vdots & \vdots & \ddots & \vdots \\ m_{r1}N & m_{r2}N & \ldots & m_{rs}N \end{pmatrix},$$

---

[1] V. Tarokh is with the Coordinated Science Laboratory, University of Illinois at Urbana-Champiagn.

[2] I.F. Blake is with the Department of Electrical Engineering, The University of Waterloo, Ontario, Canada.

where $m_{ij}N$ is the product of the scalar quantity m_{ij} and the matrix N.

Since any vector is also a matrix, the tensor product of vectors is also given by the above definition.

Definition 2 *Let L_1 and L_2 be lattices with generator matrices M_1 and M_2. The tensor product of L_1 and L_2 denoted by $L_1 \otimes L_2$ is defined by the lattice generated by the rows of $M_1 \otimes M_2$.*

Note that generator matrices of L_1 and L_2 are not unique. Nevertheless, all the lattices formed from the tensor products of arbitrary generator matrices of L_1 and L_2 are equivalent [1], so that Definition 2 makes sense.

For $L \in \mathcal{L}$, let $d(L)$ denotes the smallest length amongst nonzero points of L. Let $\mathbb{Z}^+$ denote the set of non-negative integers.

The following lemma is an easy but fundamental observation:

Lemma 1 *For any lattices L_1 and L_2,*

$$d(L_1 \otimes L_2) \leq d(L_1)d(L_2). \tag{2}$$

Proof: Let $v_1 \in L_1$ and $v_2 \in L_2$ have Euclidean lengths $d(L_1)$ and $d(L_2)$, respectively. It is easy to see that $v_1 \otimes v_2$ is a vector in $L_1 \otimes L_2$. By direct computation, $v_1 \otimes v_2$ has length $d(L_1)d(L_2)$. Thus the minimum length of $L_1 \otimes L_2$ is certainly less than or equal to this number. This completes the proof. □

Motivated by the inequality 2, we define the class $\mathcal{TE}$ to consist of all lattices L such that $d(L \otimes L') = d(L)d(L')$, for an arbitrary lattice L'. It is clear that the lattices $\mathbb{Z}^n \in \mathcal{TE}$ for all $n \in \mathbb{N}$, thus $\mathcal{TE} \neq \emptyset$.

The class $\mathcal{TE}$ has been studied extensively by many mathematicians including Steinberg and Kitaoka. The results that will be used here are mostly due to Kitaoka ([2], [3],[4], [5]). In particular Kitaoka proved that:

Theorem 1 Kitaoka's First Theorem: *Any lattice of dimension ≤ 42 is necessarily in $\mathcal{TE}$.*

Theorem 2 Kitaoka's Second Theorem: *Let L be a lattice with $d(L) \leq 6$ such that any nonzero sublattice L_0 of L has determinant ≥ 1. Then $L \in \mathcal{TE}$.*

Corollary 1 *$D_n \in \mathcal{TE}$ for all $n \geq 4$.*

The proofs of these theorems are sophisticated and irrelevant to the topic of this work. The interested reader is referred to [2], [3], [4], [5].

In the following, Kitaoka's theorems are used to establish upper bounds on $\mathcal{T}_3$. The following lemmas are established in [6].

Lemma 2 *If $L_1, L_2 \in \mathcal{L}$, then $L_1 \otimes L_2 \in \mathcal{L}$.*

Lemma 3 *For L_1 and $L_2 \in \mathcal{L}$, we have:*

$$\mathcal{G}(L_1 \otimes L_2) \leq \mathcal{G}(L_1)\mathcal{G}(L_2). \tag{3}$$

Lemma 4 *Let $L \in \mathcal{L}$ be of dimension m. Then*

$$V(D_n \otimes L) = 2^m V(L)^n. \tag{4}$$

Remark: If L_1 and L_2 are lattices of dimensions n and m respectively, it is easy to see that

$$V(L_1 \otimes L_2) = V(L_1)^m V(L_2)^n.$$

Furthermore, if L_1 or $L_2 \in \mathcal{TE}$, then

$$\delta(L_1 \otimes L_2) = \delta(L_1)\delta(L_2).$$

The above equation and Kitaoka's theorems can be used to construct lattices with arbitrarily large coding gains.

Theorem 3 *For any $\gamma \geq 1$, we have $\mathcal{T}_3(2\gamma) \leq 2\mathcal{T}_3(\gamma)$.*

Proof: For $\beta \geq 1$, let $L \in \mathcal{L}$ be an arbitrary lattice such that $\delta(L) \geq \beta$. It follows from the corollary to Kitaoka's Second Theorem that $d(D_n \otimes L) = d(D_n)d(L) = \sqrt{2}d(L)$ for all $n \geq 4$. By Lemma 4, assuming that L has dimension m, $V(D_n \otimes L) = 2^m V(L)^n$. Using these results, a simple computation shows that:

$$\delta(D_n \otimes L) = 2^{(n-2)/n}\delta(L) \geq 2^{(n-2)/n}\beta. \tag{5}$$

Apply Lemma 3 now to obtain $\mathcal{G}(D_n \otimes L) \leq \mathcal{G}(D_n)\mathcal{G}(L)$. Since $\mathcal{G}(D_n) \leq 2$, we conclude that

$$\mathcal{G}(D_n \otimes L) \leq 2\mathcal{G}(L). \tag{6}$$

It follows from 5, 6 and the definition of $\mathcal{T}_3$ that

$$\mathcal{T}_3(2^{(n-2)/n}\beta) \leq 2\mathcal{G}(L), \tag{7}$$

for all $L \in \mathcal{L}$ such that $\delta(L) \geq \beta$. Take the infimum over all such L to obtain

$$\mathcal{T}_3(2^{(n-2)/n}\beta) \leq 2\mathcal{T}_3(\beta), \tag{8}$$

for all $n \geq 4$. Let $n \to \infty$. By the continuity of $\mathcal{T}_3$, we have

$$\lim_{n\to\infty} \mathcal{T}_3(2^{(n-2)/n}\beta) = \mathcal{T}_3(2\beta).$$

We thus obtain $\mathcal{T}_3(2\beta) \leq 2\mathcal{T}_3(\beta)$, concluding the proof. □

Theorem 4 *For any $\gamma \geq 1$, we have*

$$\mathcal{T}_3(\gamma) \leq 2\gamma. \tag{9}$$

Proof: By Theorem 3 we have $\mathcal{T}_3(2\gamma) \leq 2\mathcal{T}_3(\gamma)$, for all $\gamma \geq 1$. In particular $\mathcal{T}_3(2) \leq 2\mathcal{T}_3(1) = 2$. Since $\mathcal{T}_3$ is increasing, we observe that $\mathcal{T}_3(\gamma) \leq 2$ for $1 \leq \gamma \leq 2$. A simple induction on the result of Theorem 3 proves that

$$\mathcal{T}_3(2^m\beta) \leq 2^m\mathcal{T}_3(\beta), \tag{10}$$

for all $\beta \geq 1$ and $m \in \mathbb{Z}^+$.

Next assume that $\gamma > 1$ is arbitrarily given, and choose $m \in \mathbb{Z}^+$ such that $2^m < \gamma \leq 2^{m+1}$. Then $2 \geq (\gamma/2^m) > 1$ and thus $\mathcal{T}_3(\gamma/2^m) \leq 2$. It follows now from 10 that

$$\mathcal{T}_3(\gamma) \leq 2^m\mathcal{T}_3(\gamma/2^m) \leq 2^{m+1} \leq 2\gamma. \tag{11}$$

This concludes the proof. □

Thus the label trellis complexity function grows at most linearly in terms of the coding gain. This demonstrates that the difficulty encountered in decoding rational lattices using the Viterbi algorithm originates from the comparison and addition operations.

References

[1] J.H. Conway and N.J.A. Sloane, *Sphere Packings, Lattices and Groups.* New York: Springer-Verlag, 1993.

[2] Y. Kitaoka, "Scalar extensions of quadratic lattices II," *Nagoya Mathematics Journal.* Vol 67, pp. 159-164. 1977.

[3] Y. Kitaoka, "Tensor products of positive definite quadratic forms III," *Nagoya Mathematics Journal.* Vol 70, pp. 173-181. 1978.

[4] Y. Kitaoka," Tensor products of positive definite quadratic forms V," *Nagoya Mathematics Journal.* Vol 82, pp. 99-111. 1981.

[5] Y. Kitaoka, "Tensor products of positive definite quadratic forms VII," *Nagoya Mathematics Journal.* Vol 96, pp. 133-137. 1984.

[6] V. Tarokh, *Trellis complexity versus the coding gain of lattice-based communication systems,* Ph.D. thesis, University of Waterloo, Waterloo, Ontario, Canada. 1995.

On codes satisfying the double chain condition [1]

Torleiv Kløve,
Department of Informatics, University of Bergen
HIB, N-5020 Bergen, Norway.

Abstract

The double chain condition is described. A number of bounds on the length and weight hierarchy of codes satisfying the double chain condition are given.

1 Introduction and notations

We consider binary linear codes. The *support* of a vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ in $GF(2)^n$ is defined by

$$\chi(\mathbf{x}) = \{i \mid x_i \neq 0\},$$

and the support of a subset $S \subseteq GF(2)^n$ is defined by

$$\chi(S) = \bigcup_{\mathbf{x} \in S} \chi(\mathbf{x}).$$

The *support weight* of S is defined by

$$w_S(S) = |\chi(S)|.$$

Hence, $w_S(S)$ is the number of positions where at least one vector in S is non-zero. The *weight hierarchy* of an $[n, k]$ code C is the sequence $(d_1, d_2, \ldots, d_k)$, where

$$d_r = d_r(C) = \min\{w_S(D) \mid D \text{ is an } [n, r] \text{ subcode of } C\}.$$

In particular, $d_1 = d$, the minimum distance of C. The parameters $d_1, d_2, \ldots, d_k$ of a code were first introduced by Helleseth et al. [4]. Forney [2] called $(d_1, d_2, \ldots, d_k)$ the *length/dimension profile* and he also considered the inverse, the *dimension/length profile* $(k_0, k_1, \ldots, k_n)$ defined by:

$$k_i = r \text{ for } d_r \leq i < d_{r+1}.$$

In particular, $k_i = 0$ for $i < d$ and $k_n = k$.

Forney [3] introduced the *double chain condition* which can be rephrased as follows. An $[n, k]$ code C is called a DCC (double chain condition) code if it has the following property: there exist two chains of subcodes of C, the *left chain*

$$D_1^{\mathrm{L}} \subset D_2^{\mathrm{L}} \subset \cdots \subset D_k^{\mathrm{L}} = C,$$

and the *right chain*

$$D_1^{\mathrm{R}} \subset D_2^{\mathrm{R}} \subset \cdots \subset D_k^{\mathrm{R}} = C,$$

where, for $1 \leq r \leq k$, we have

$$\dim(D_r^{\mathrm{L}}) = \dim(D_r^{\mathrm{R}}) = r,$$

$$\chi(D_r^{\mathrm{L}}) = \{1, 2, \ldots, d_r\},$$

---

[1] The research for this paper was supported by The Norwegian Research Council, grants no. 107542/410 and 107623/420.

$$\chi(D_r^{\mathrm{R}}) = \{n - d_r + 1, n - d_r + 2, \ldots, n\}.$$

A code is said to satisfy the double chain condition if it is equivalent to a DCC code. Kasami et al. [8] showed that the Reed-Muller codes satisfy the double chain condition, and Forney [2], [3] proved that several other classes of codes have this property.

Forney [2] defined the *state complexity profile* $(s_0, s_1, \ldots, s_n)$ of an $[n, k]$ code and gave a lower bound on the s_i in terms of the dimension/length profile and what he called the *inverse* dimension/length profile. Codes satisfying the double chain condition are optimal with respect to this bound in the sense that the bound is satisfied with equality for all i, and this is our reason to studying these codes. For these codes the s_i are given by

$$s_i = k - k_i - k_{n-i}$$

for $0 \leq i \leq k$. Further, the *state complexity* is

$$s = \max\{s_i \mid 0 \leq i \leq n\}.$$

Sometimes we will include s and d in the notation for an $[n, k]$ code C, and refer to C as an $[n, k, d]$ and $[n, k, s, d]$ code. Further, if C is a DCC code, we will also refer to it as an $[n, k]^{\mathrm{DCC}}$, $[n, k, d]^{\mathrm{DCC}}$, and $[n, k, s, d]^{\mathrm{DCC}}$ code.

In this paper we give some general bounds on the parameters of DCC codes. We also describe the parameters n, k, d for which there exist $[n, k, 1, d]^{\mathrm{DCC}}$ and $[n, k, 2, d]^{\mathrm{DCC}}$ codes.

An $[n, k]^{\mathrm{DCC}}$ code C has a basis $\mathcal{G} = \{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_k\}$ such that

$$\langle \mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_r \rangle = D_r^{\mathrm{L}} \quad \text{for} \quad 1 \leq r \leq k. \tag{1}$$

Here $\langle \mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_r \rangle$ denotes the vector space spanned by $\{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_r\}$. Similarly, for a vector space D and a vector $\mathbf{x}$ we will use the notation $\langle D, \mathbf{x} \rangle$ to denote the space spanned by D and $\mathbf{x}$, etc. In the following, when we consider an $[n, k]^{\mathrm{DCC}}$ code C we will assume that a basis $\mathcal{G}$ has been chosen such that (1) is satisfied. We note that such a basis is not unique since we may substitute $\mathbf{g}_i + \sum_{j=1}^{i-1} \alpha_j \mathbf{g}_j$ for $\mathbf{g}_i$ without effecting (1). We as usual write $\mathbf{g}_r = (g_{r1}, g_{r2}, \ldots, g_{rn})$, and we will refer to these elements without further comments. We note that

$$\begin{aligned} g_{rd_r} &= 1 \\ g_{ri} &= 0 \qquad \text{for} \quad d_r < i \leq n. \end{aligned}$$

Similarly, C has a basis $\mathcal{H} = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_k\}$ such that

$$\langle \mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_r \rangle = D_r^{\mathrm{R}} \text{ for } \quad 1 \leq r \leq k. \tag{2}$$

For any vector $\mathbf{x} \in C \setminus \{\mathbf{0}\}$, let

$$l(\mathbf{x}) = \min \chi(\mathbf{x}) \quad \text{and} \quad u(\mathbf{x}) = \max \chi(\mathbf{x}),$$

that is, $l(\mathbf{x})$ and $u(\mathbf{x})$ are the positions of the leftmost and rightmost 1 in $\mathbf{x}$ respectively.

Lemma 1 *Let C be an $[n, k, d]^{\mathrm{DCC}}$ code. For all $\mathbf{x} \in C \setminus \{\mathbf{0}\}$ we have*

$$\begin{aligned} &(i) \quad u(\mathbf{x}) = d_r \qquad \textit{for some } r, \\ \textit{and} \quad &(ii) \quad l(\mathbf{x}) = n + 1 - d_{r'} \quad \textit{for some } r'. \end{aligned}$$

Proof: Since $\mathcal{G}$ is a basis, there exist $a_1, a_2, \ldots, a_r$ for some r, $1 \le r \le k$ such that

$$\mathbf{x} = \sum_{i=1}^{r} a_i \mathbf{g}_i,$$

and $a_r = 1$. By the definition of the chain condition, we have

$$g_{ij} = 0 \quad \text{if} \quad 1 \le i \le r \quad \text{and} \quad d_r < j \le n,$$

$$g_{id_r} = 0 \quad \text{if} \quad 1 \le i < r,$$

and

$$g_{rd_r} = 1.$$

Hence $u(\mathbf{x}) = d_r$. A similar argument, using the basis $\mathcal{H}$ gives *(ii)*. □

Corollary 1 *Let C be an $[n, k, d]^{\mathrm{DCC}}$ code. Then there exists a basis $\mathcal{G}$ and a permutation π of $\{1, 2, \ldots, k\}$ such that*

$$\langle \mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_r \rangle = D_r^{\mathrm{L}} \quad \textit{for} \quad 1 \le r \le k,$$

and

$$\langle \mathbf{g}_{\pi(1)}, \mathbf{g}_{\pi(2)}, \ldots, \mathbf{g}_{\pi(r)} \rangle = D_r^{\mathrm{R}} \quad \textit{for} \quad 1 \le r \le k.$$

That is, we can choose $\mathcal{H}$ as a permutation of $\mathcal{G}$.

Proof: Let $\mathcal{G}$ be a basis for C satisfying (1). If $i < j$ are such that $l(\mathbf{g}_i) = l(\mathbf{g}_j)$, then we can replace $\mathbf{g}_j$ by $\mathbf{g}_i + \mathbf{g}_j$. This will not effect the property (1). Repeating these substitutions if necessary, we see that we may assume that $l(\mathbf{g}_i) \ne l(\mathbf{g}_j)$ for all $i \ne j$. From Lemma 1(ii) we see that

$$\{l(\mathbf{g}_r) \mid 1 \le r \le k\} = \{n + 1 - d_{r'} \mid 1 \le r' \le k\},$$

and the corollary follows.

In a vector or matrix, a block of a consecutive zeros will sometimes be denoted by $\overbrace{0}^{a}$, and similarly for a block of ones.

2 Some basic results

Theorem 1 *If C is an $[n, k]^{\mathrm{DCC}}$ code, then $d_k = n$.*

Proof: Suppose that $d_k < n$. By the left chain condition $n \notin \chi(C)$. By the right chain condition $n \in \chi(C)$, a contradiction. □

Lemma 2 *If C is an $[n, k, d]$ code with $k > 2$ which contains two codewords*

$$\mathbf{x} = (\overbrace{1}^{a}\ \overbrace{1}^{b}\ \overbrace{0}^{a}) \quad \textit{and} \quad \mathbf{y} = (\overbrace{0}^{a}\ \overbrace{1}^{b}\ \overbrace{1}^{a}),$$

where $a + b = d$, then $b = 0$.

Proof: Write the codewords $\mathbf{c}$ of C as

$$\mathbf{c} = (\mathbf{c}_1|\mathbf{c}_2|\mathbf{c}_3)$$

where $\mathbf{c}_1$ and $\mathbf{c}_3$ have length a and $\mathbf{c}_2$ has length b. Let

$$\mathbf{z} = \mathbf{x} + \mathbf{y} = (\mathbf{1}|\mathbf{0}|\mathbf{1}).$$

For any codeword $\mathbf{c}$, we have $\mathbf{c} + \mathbf{z} \in C$. If $\mathbf{c} \notin \{\mathbf{0}, \mathbf{z}\}$ we have

$$\begin{aligned} 2a + 2b &= d + d \\ &\leq w(\mathbf{c}) + w(\mathbf{c} + \mathbf{z}) \\ &= \Big(w(\mathbf{c}_1) + w(\mathbf{c}_1 + \mathbf{1})\Big) + \Big(w(\mathbf{c}_2) + w(\mathbf{c}_2)\Big) \\ &\quad + \Big(w(\mathbf{c}_3) + w(\mathbf{c}_3 + \mathbf{1})\Big) \\ &= a + 2w(\mathbf{c}_2) + a \\ &\leq 2a + 2b \end{aligned}$$

since $w(\mathbf{c}_2) \leq b$. Hence $w(\mathbf{c}_2) = b$ (and $\mathbf{c}_2 = \mathbf{1}$). Let $\tilde{\mathbf{c}}$ be a codeword in C, not in $\{\mathbf{0}, \mathbf{x}, \mathbf{y}, \mathbf{z}\}$, and let

$$\mathbf{c} = \tilde{\mathbf{c}} + \mathbf{x} = ((\tilde{\mathbf{c}}_1 + \mathbf{1})|\mathbf{0}|\tilde{\mathbf{c}}_3).$$

Then $b = w(\mathbf{c}_2) = w(\mathbf{0}) = 0$. ☐

Theorem 2 *If C is an $[n, k > 2, d]^{\mathrm{DCC}}$ code, then $2d \leq n$.*

Proof: Let $D_1^{\mathrm{L}} = \{\mathbf{0}, \mathbf{x}\}$ and $D_1^{\mathrm{R}} = \{\mathbf{0}, \mathbf{y}\}$. By Lemma 2, $\chi(\mathbf{x}) \cap \chi(\mathbf{y}) = \emptyset$ and so

$$2d = |\chi(\mathbf{x})| + |\chi(\mathbf{y})| = |\chi(\mathbf{x}) \cup \chi(\mathbf{y})| \leq n.$$

☐

Example. The simplex codes have parameters $[2^m - 1, m, 2^{m-1}]$. By Theorem 2, the simplex codes do not satisfy the double chain condition. In contrast, Kasami et al. [8] showed that the closely related $[2^m, m+1, 2^{m-1}]$ first order Reed-Muller codes do satisfy the double chain condition for all m.

Theorem 3 *If C is an $[n, k, d]^{\mathrm{DCC}}$ code, then*

$$d_{r+1} \leq d_r + d$$

for $1 \leq r < k$. In particular $d_r \leq rd$ for all r and $n \leq kd$.

Proof: Let $1 \leq r < k$ and let $D = \langle D_r^{\mathrm{L}}, D_1^{\mathrm{R}} \rangle$. Since $n \notin \chi(D_r^{\mathrm{L}})$ and $n \in \chi(D_1^{\mathrm{R}})$, we have $\dim(D) = r + 1$. Hence

$$d_{r+1} \leq w_S(D) \leq w_S(D_r^{\mathrm{L}}) + w_S(D_1^{\mathrm{R}}) = d_r + d.$$

☐

In [11], Lafourcade and Vardy proved that for any $[n, k, s, d]$ code we have

$$n \geq \frac{k}{s}(d-1). \tag{3}$$

For codes satisfying the double chain condition we can give stronger bounds on n. We will also give bounds on d_r in general.

By Theorem 3, if $d = 1$ for an $[n, k]^{\mathrm{DCC}}$ code C, then $n = k$ and so $C = GF(2)^n$. Further, the only $[kd, k, d]^{\mathrm{DCC}}$ codes are the $[kd, k, 0, d]^{\mathrm{DCC}}$ codes generated by the matrices

$$\begin{pmatrix} \overbrace{1}^{d} & \overbrace{0}^{d} & \cdots & \overbrace{0}^{d} \\ 0 & 1 & \cdots & 0 \\ \cdots & & & \\ 0 & 0 & \cdots & 1 \end{pmatrix}.$$

Therefore, from now on we will assume that $s \geq 1$, $d \geq 2$, and $n < kd$.

Lemma 3 *For an $[n, k, s, d \geq 2]^{\mathrm{DCC}}$ code we have*

$$d_r + d_{k-r-s+1} \leq n + 1.$$

Proof: Let $i = d_r - 1$. By definition, $k_i = r - 1$ and

$$k_{n-i} = k - k_i - s_i \geq k - r + 1 - s$$

and so

$$n - i \geq d_{k-r+1-s}$$

and

$$n \geq d_r - 1 + d_{k-r+1-s}.$$

Corollary 2 *Let C be an $[n, k, s, d \geq 2]^{\mathrm{DCC}}$ code. If $r + t \leq k - s$, then*

$$dim(\langle D_r^{\mathrm{L}}, D_t^{\mathrm{R}} \rangle) = r + t$$

and

$$w_S(\langle D_r^{\mathrm{L}}, D_t^{\mathrm{R}} \rangle) = d_r + d_t.$$

Corollary 3 *Let C be an $[n, k, s, d \geq 2]^{\mathrm{DCC}}$ code. If $r + t = k - s + 1$, then*

$$dim(\langle D_r^{\mathrm{L}}, D_t^{\mathrm{R}} \rangle) = r + t$$

and

$$d_r + d_t - 1 \leq w_S(\langle D_r^{\mathrm{L}}, D_t^{\mathrm{R}} \rangle) \leq d_r + d_t.$$

Proof: If $r + t \leq k - s$, then, by Lemma 3,

$$d_r + d_t \leq d_r + d_{t+1} - 1 \leq d_r + d_{k-r-s+1} - 1 \leq n$$

and so $\chi(D_r^{\mathrm{L}}) \cap \chi(D_t^{\mathrm{R}}) = \emptyset$. Hence

$$dim(\langle D_r^{\mathrm{L}}, D_t^{\mathrm{R}} \rangle) = r + t$$

and

$$w_S(\langle D_r^{\mathrm{L}}, D_t^{\mathrm{R}}\rangle) = d_r + d_t.$$

If $r + t = k - s + 1$ we get in the same way that

$$d_r + d_t - 1 \le w_S(\langle D_r^{\mathrm{L}}, D_t^{\mathrm{R}}\rangle).$$

Assume that $dim(\langle D_r^{\mathrm{L}}, D_t^{\mathrm{R}}\rangle) < r + t$. This is only possible if $\mathbf{g}_r \in D_t^{\mathrm{R}}$ and so $l(\mathbf{g}_r) \ge n + 1 - d_t \ge d_r$. Hence $l(\mathbf{g}_r) = d_r = u(\mathbf{g}_r)$ and $w_{\mathrm{H}}(\mathbf{g}_r) = 1 < d$, a contradiction. □

Theorem 4 *For an* $[n, k, s, d \ge 2]^{\mathrm{DCC}}$ *code we have*

$$d_{r+s+t-1} \ge d_r + d_t - 1$$

for $r \ge 1$, $t \ge 1$, *and* $r + s + t - 1 \le k$.

Proof: Let

$$D = \langle D_r^{\mathrm{L}}, D_{k-r-s+1}^{\mathrm{R}}\rangle.$$

By Corollary 2, $\dim(D) = k - s + 1$. Since the vectors $\mathbf{g}_{r+1}, \mathbf{g}_{r+2}, \dots, \mathbf{g}_{r+s+t-1}$ are linearly independent, and

$$\dim(\langle D, \mathbf{g}_{r+1}, \mathbf{g}_{r+2}, \dots, \mathbf{g}_{r+s+t-1}\rangle) \le k,$$

there exist $i_1, i_2, \dots, i_t$ such that

$$r + 1 \le i_1 < i_2 < \dots < i_t \le r + s + t - 1$$

and

$$\mathbf{g}_{i_u} \in D \quad \text{for} \quad 1 \le u \le t,$$

that is

$$g_{i_u} = \mathbf{y}_u + \mathbf{z}_u$$

where $\mathbf{y}_u \in D_r^{\mathrm{L}}$ and $\mathbf{z}_u \in D_{k-r-s+1}^{\mathrm{R}}$. Suppose

$$\sum_{u=1}^{t} a_u \mathbf{z}_u = 0$$

for some $a_u \in GF(2)$. Then

$$\sum_{u=1}^{t} a_u \mathbf{g}_{i_u} = \sum_{u=1}^{t} a_u \mathbf{y}_u \in D_r^{\mathrm{L}}$$

and so $a_u = 0$ for all u; that is, the vectors $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_t$ are linearly independent. Let

$$D' = \langle \mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_t\rangle.$$

Then

$$\max \chi(D') = d_{i_t} \le d_{r+s+t-1}, \tag{4}$$

and

$$\begin{aligned} \min \chi(D') &\ge \min \chi(D_{k-r-s+1}^{\mathrm{R}}) \\ &= n + 1 - d_{k-r-s+1} \ge d_r \end{aligned} \tag{5}$$

by Lemma 3. Combining (4) and (5) we get

$$d_{r+s+t-1} \geq d_r + w_S(D') - 1 \geq d_r + d_t - 1.$$

□

Let

$$g(r,d) = \sum_{i=0}^{r-1} \left\lceil \frac{d}{2^i} \right\rceil$$

denote the Griesmer bound. It is well known that

$$d_r \geq g(r,d).$$

Theorem 5 *For an* $[n,k,s,d \geq 2]^{\mathrm{DCC}}$ *code* C, *for* $t \geq 1$, *and for* $1 \leq r \leq k$, *write*

$$r = a(s+t-1) + b$$

where $2 \leq b \leq s+t$. *Further, let*

$$b_1 = \lfloor b/2 \rfloor \quad \text{and} \quad b_2 = b - b_1.$$

Then

$$d_r \geq a\left(g(t,d) - 1\right) + g(b_1,d) + g(b_2,d) - 1.$$

Proof: By Theorem 4 we get

$$d_{b_1+i(s+t-1)} \geq d_{b_1} + i(d_t - 1)$$

for $i = 0, 1, \ldots, a$, and

$$\begin{aligned} d_r &\geq d_{b_1+a(s+t-1)} + d_{b_2} - 1 \\ &\geq a(d_t - 1) + d_{b_1} + d_{b_2} - 1 \\ &\geq a\left(g(t,d) - 1\right) + g(b_1,d) + g(b_2,d) - 1. \end{aligned}$$

□

Example. If d is even and $k = a(s+1) + 2$ for some integer a, we can choose $t = 2$, $b_1 = b_2 = 1$ in the theorem and get

$$n \geq \frac{k-2}{s+1}\left(\frac{3}{2}d - 1\right) + 2d - 1,$$

compared to Lafourcade and Vardy general bound (3):

$$n \geq \frac{k}{s}(d-1).$$

E.g. for $s = 10$, $d = 2$, $k = 90 = 8(10+1) + 2$ we get $n \geq 23$ compared to $n \geq 10$.

3 Codes with trellis complexity one

Theorem 6 *For an* $[n,k,1,d\geq 2]^{\mathrm{DCC}}$ *code we have*

$$\begin{aligned}
&a)\ d_{r+1}\geq d_r+d-1 && \text{for}\quad 1\leq r<k,\\
&b)\ d_r\geq r(d-1)+1 && \text{for}\quad 1\leq r<k,\\
&c)\ n\geq k(d-1)+1.
\end{aligned}$$

Proof: We see that a) follows directly from Theorem 4 and that b) follows from a) by induction. Finally, c) follows from b) and Theorem 1, or alternatively, by putting $s=t=1$ in Theorem 5. □

By Theorems 3 and 6, for an $[n,k,1,d\geq 2]^{\mathrm{DCC}}$ it is necessary that $dk-k+1\leq n<dk$. The main result of this section is to show that this is also sufficient, i.e. for all such n there do exist $[n,k,1,d]^{\mathrm{DCC}}$ codes. We do this by giving explicit code constructions of $[n,k,1,d\geq 2]^{\mathrm{DCC}}$ codes for all n, k, d for which $dk-k+1\leq n<dk$. To save space, we only sketch the proof.

To give a compact description of the codes we will present, we introduce another notation. To a sequence $(b_0,a_1,b_1,a_2,b_2,\ldots,b_{k-1},a_k,b_k)$ of non-negative integers we assosiate a generator matrix

$$\begin{array}{ccccccccc}
\overbrace{}^{b_0} & \overbrace{}^{a_1} & \overbrace{}^{b_1} & \overbrace{}^{a_2} & \overbrace{}^{b_2} & & \overbrace{}^{b_{k-1}} & \overbrace{}^{a_k} & \overbrace{}^{b_k}
\end{array}$$
$$\left(\begin{array}{ccccccccc}
1 & 1 & 1 & 0 & 0 & \cdots & 0 & 0 & 0\\
0 & 0 & 1 & 1 & 1 & \cdots & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 1 & \cdots & 0 & 0 & 0\\
\cdots & & & & & & & & \\
0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & \cdots & 1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & \cdots & 1 & 1 & 1
\end{array}\right)$$

of an $[n,k,d]$ code $C(b_0a_1b_1a_2b_2\cdots b_{k-1}a_kb_k)$ where

$$n=\sum_{i=1}^{k}a_i+\sum_{i=0}^{k}b_i.$$

If

$$\begin{aligned}
&a_r=a_{k+1-r} && \text{for}\quad 1\leq r\leq k,\\
&b_r=b_{k-r} && \text{for}\quad 0\leq r\leq k,\\
&b_0+a_1+b_1=d,\\
&a_r+b_r\leq d && \text{for}\quad 1\leq r\leq k,\\
&b_{r-1}+a_r+b_r\geq d && \text{for}\quad 1\leq r\leq k,
\end{aligned}$$

we call such a code an *ab*-code. If in addition

$$b_r\in\{0,1\}\quad \text{for}\quad 0\leq r\leq k,$$

we call the code a 1-*ab*-code. Note that this implies that

$$a_r\in\{d-2,d-1,d\}\quad \text{for}\quad 1\leq r\leq k.$$

For a sequence $\bar{a}=(a_1,a_2,\ldots,a_k)$ define

$$\sigma(u,j)=\sigma(\bar{a};u,j)=\sum_{i=u}^{u+j-1}a_i.$$

We can prove the following sufficient condition.

Lemma 4 *Let* $(a_1, a_2, \ldots, a_m)$ *be a sequence such that* $a_i = a_{m+1-i}$ *for all* i, *and* $|\sigma(u, j) - \sigma(u', j)| \leq 1$ *for all* u, u', j *such that* $1 \leq j \leq m$ *and* $1 \leq u \leq u' \leq m - j + 1$. *Then the 1-ab-codes* C_t *defined by*

$$C_t = C\Big(1a_1 1a_2 1 \ldots 1a_{m-1} 1 (a'_m 0 a'_1 1a_2 1 \ldots 1a_{m-1} 1)^t a_m 1\Big),$$

where $a'_1 = a_1 + 1$ *and* $a'_m = a_m + 1$, *is a DCC code for all* $t \geq 0$.

Theorem 7 $[n, k, 1, d \geq 2]^{\mathrm{DCC}}$ *codes exist if and only if* $dk - k + 1 \leq n < dk$.

Proof: By Theorems 3 and Corollary 6, for an $[n, k, 1, d \geq 2]^{\mathrm{DCC}}$ it is necessary that $dk - k + 1 \leq n < dk$. It remains to show the if part, and we do this by giving explicit constructions. We use the notation $\{x\}$ for the integer closest to x, with the special case $\{n + 0.5\} = n$ for all integers n.
Case I, $n = kd - 2p - 1$ where $0 \leq p \leq (k-2)/2$. Use

$$\begin{aligned} a_r &= d - 2 \quad \text{for} \quad r = \left\{\tfrac{k-1}{2p+1} i + 1\right\},\ 0 \leq i \leq 2p+1, \\ a_r &= d - 1 \quad \text{otherwise,} \end{aligned}$$

and $t = 0$ in Lemma 4.
Case II, k is odd and $n = kd - 2p$, where $0 < p \leq (k-1)/2$. Use

$$\begin{aligned} a_r = a_{k+1-r} &= d - 2 \quad \text{for} \quad r = \left\{\tfrac{k-1}{2p} i + 1\right\},\ 0 \leq i \leq p, \\ a_r &= d - 1 \quad \text{otherwise,} \end{aligned}$$

and $t = 0$ in Lemma 4.
Case III, k is even and $n = kd - 2p$, where $0 < p \leq m - 1$. Let $k = \alpha m$, where α is even and m is odd. Use the construction in cases I and II (with m substituted for k) and $t = \alpha - 1$ in Lemma 4.

4 Codes with trellis complexity two

We now consider the parameters n, k, d for which $[n, k, 2, d \geq 2]^{\mathrm{DCC}}$ codes exist. Since $[n, k, 1, d \geq 2]^{\mathrm{DCC}}$ codes exist for $n > kd - k$, we restrict our attention to $n \leq kd - k$. We can prove The following results:

- For even d, $[n, k, 2, d \geq 2]^{\mathrm{DCC}}$ codes exist if and only if $n \geq (k+1)d/2$.
- For odd d $[n, k, 2, d \geq 2]^{\mathrm{DCC}}$ codes exist for $n \geq (k+1)(d-1)/2 + k$. We believe that no $[n, k, 2, d \geq 2]^{\mathrm{DCC}}$ codes exist for $n < (k+1)(d-1)/2 + k$, but we can only show a slightly weaker result:

  For an $[n, k, 2, d \geq 5]^{\mathrm{DCC}}$ code, where d is odd, we have

  $$d_r \geq (r+1)\frac{d}{2} - \frac{r - \beta_r}{10} \text{ for } 1 \leq r \leq k,$$

  where

  $$\beta_r = \begin{cases} 5 & \text{for } r \equiv 0 \pmod 5, \\ 1 & \text{for } r \equiv 1 \pmod 5, \\ -3 & \text{for } r \equiv 2 \pmod 5, \\ 3 & \text{for } r \equiv 3 \pmod 5, \\ -1 & \text{for } r \equiv 4 \pmod 5. \end{cases}$$

- A computer search showed that there are no $[n,k,2,2\delta+1]^{\mathrm{DCC}}$ codes with $\delta > \frac{k+5}{3}$ and $n < (k+1)\delta + k$ for $k \leq 12$. If there exist any $[n,k,2,2\delta+1]^{\mathrm{DCC}}$ codes with $n < (k+1)\delta + k$ at all is an open question.

References

[1] S. Encheva and T. Kløve, Codes satisfying the chain condition, IEEE Trans. Inform. Theory 40 (1994) 175-180.

[2] G. D. Forney, Dimension/length profiles and trellis complexity of linear block codes, IEEE Trans. Inform. Theory 40 (1994) 1741-1752.

[3] G. D. Forney, Dimension/length profiles and trellis complexity of lattices, IEEE Trans. Inform. Theory 40 (1994) 1753-1772.

[4] T. Helleseth, T. Kløve, J. Mykkeltveit, The weight distribution of irreducible cyclic codes, Discrete Math. 18 (1977) 179-211.

[5] T. Helleseth, T. Kløve, Ø. Ytrehus, Generalized Hamming weights of linear codes, IEEE Trans. Inform. Theory 38 (1992) 1133-1140.

[6] T. Helleseth, T. Kløve, Ø. Ytrehus, Codes and the chain condition, Proc. Int. workshop on Algebraic and Combinatorial Coding Theory, Voneshta Voda, Bulgaria June 22-28 (1992) 88-91.

[7] T. Helleseth, T. Kløve, Ø. Ytrehus, Codes, weight hierarchies, and chains, Proc. ICCS/ISITA '92, Singapore, 16-20 November (1992) 608-612.

[8] T. Kasami, T. Takata, T. Fujiwara, S. Lin, On the optimum bit orders with respect to the state complexity of trellis diagrams for binary linear codes, IEEE Trans. Inform. Theory 39 (1993) 242-245.

[9] T. Kløve, Support weight distribution of linear codes, Discrete Math. 106/107 (1992) 311-316.

[10] T. Kløve, Minimum support weights of binary codes, IEEE Trans. Inform. Theory 39 (1993) 648-654.

[11] A. Lafourcade and A. Vardy, Asymptotically good codes have infinite trellis complexity, IEEE Trans. Inform. Theory 41 (1995) 555-559.

[12] V. K. Wei, Generalized Hamming Weights for Linear Codes, IEEE Trans. Inform. Theory 37 (1991) 1412-1418.

FAIR BLOCKING MECHANISMS FOR DISCRETE-TIME MULTIPLEXING

Adrian E. Conway,[1] GTE Laboratories Incorporated, Waltham, MA, aec0@gte.com
Catherine Rosenberg,[2] Ecole Polytechnique, Montreal, Canada, cath@comm.polymtl.ca

Extended Abstract*: In future integrated-service packet networks, it is envisaged that a range of qualities of service will be offered to applications with different requirements. To achieve economies of scale, extensive use must be made of statistical multiplexing. A major challenge is, therefore, to devise multiplexing systems that can support services with different QoS requirements in terms of loss and delay. In this paper, we focus on the problem of satisfying different loss requirements in a packet multiplexer. Many schemes have, of course, already been proposed to deal with this problem. For example, there are selective discard mechanisms such as Partial Buffer Sharing (PBS) and Push-Out. One can also make use of different buffer allocation policies, such as Complete Partitioning or Complete Sharing (CS), and dimension them appropriately to satisfy loss requirements.

The fundamental problem of satisfying different loss requirements is, more generally, to offer sources a range of blocking probabilities under different scheduling disciplines. Let B_r denote the blocking probability for source r, and let ω_r be a real positive weight. The general problem is then to ensure that $B_r/\omega_r = B_1/\omega_1 = C$ for each source r, where C is some positive constant. In this paper, we introduce a mechanism that can be used to set blocking probabilities in the general ratios $B_r/B_1 = \omega_r/\omega_1$ independently of the value of the constant C. This allows for the decoupling of the buffer dimensioning problem from the parameterization of the discard mechanism, since the mechanism itself ensures the required ratios and it then remains to size the buffer to meet the required value of C. Our developments are made in the context of a time-slotted system, where it is appropriate to adopt a discrete-time view of multiplexing. This is the case in networks based on fixed-length packets, such as ATM. In each time slot, there may be a number of packet arrivals (a batch), and packets must be rejected if the batch size is greater than the number of available buffers.

A number of straightforward methods may be used for selecting the packets in a batch to be rejected. A phenomenon that we observe is that if the traffic sources are not homogeneous and the selection is made at random, this being one possible natural supposition, then the blocking probabilities seen by the sources will be imbalanced. By necessity, one method or another must be used for selecting the packets to be rejected. Instead of adopting some arbitrary method, it is preferable to reject packets in an *intelligent* manner. The mechanism that we propose is a simple and intelligent probabilistic approach for selecting packets in order to meet some loss requirements. We can distinguish two uses for the mechanism. It may be employed as a *selective discard mechanism* to provide different QoS levels in terms of loss. In this context, we call it a *Weighted Fair Blocking* (WFB) mechanism. In the special case where $\omega_r = \omega_1$ for all sources r, we achieve *fairness* in terms of blocking by forcing the blocking probabilities to be all equal. In the context of this use, we refer to the mechanism as a *Balanced Fair Blocking* (BFB) mechanism.

---

[1] Work carried out at Laboratoire MASI while on sabbatical leave from GTE Laboratories Incorporated as an Invited Professor at Universite Pierre et Marie Curie, Paris, France.

[2] Work carried out at Laboratoire MASI and at Ecole Nationale Superieure des Telecommunications, Paris, France, while on sabbatical leave from Ecole Polytechnique.

*Full-length paper available upon request (A. Conway and C. Rosenberg, Weighted Fair Blocking Mechanisms for Discrete-Time Multiplexing, *GTE Laboratories Technical Memorandum,* 1995).

The proposed mechanism offers several advantages relative to existing approaches: (1) it is relatively simple since it only requires that certain predefined probabilistic selections be made in rejecting packets, (2) it decouples the buffer dimensioning problem from the problem of parameterizing the discard mechanism, (3) it is relatively universal in the sense that it may be used on top of various multiplexing schemes such as PBS or CS, and (4) it may reduce significantly the buffer space needed to meet loss requirements. As an illustration of remark (4), we can compare the buffer requirement using our mechanism with that using random selection for a multiplexing scenario with a CS buffer allocation policy and 2 two-state MMBP sources. If the arrival rates of the sources are not equal and we use random selection, then the blocking probabilities will be imbalanced and the buffer must be dimensioned to satisfy the higher blocking probability. Let π_{ri} denote the packet arrival probability of source r when the MMBP is in state i, let $t_{xy}^{(r)}$ be the transition probability from state 'x' to 'y' in source r, and let λ_r be the mean arrival rate of source r. With $\pi_{11} = \pi_{21} = 0.0$, $\pi_{12} = \pi_{22} = 1.0$, $t_{12}^{(1)} = 0.0005$, $t_{21}^{(1)} = 0.094$, $t_{12}^{(2)} = 0.0001$, and $t_{21}^{(2)} = 0.0004$, we have $\lambda_1 = 0.005291$ and $\lambda_2 = 0.2$. In Fig. 1, we plot the blocking probabilities B_1 and B_2 for the sources as a function of the buffer size K in the case of random blocking and BFB. At K = 100, the blocking probabilities with BFB are equal to 7.27e-5. In order to achieve blocking of 7.27e-5 using random blocking, we need about 170 buffers instead of 100 buffers for BFB. Hence, at a blocking requirement of 7.27e-5, the buffer requirement is reduced by about 41% with BFB.

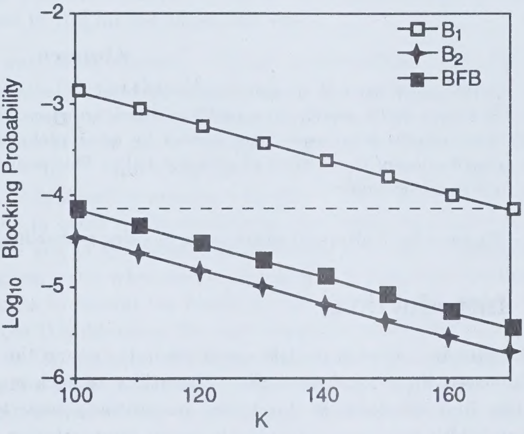


Fig. 1. Log_{10} of blocking probabilities B_1 and B_2 for random blocking and BFB as a function of buffer size K with 2 MMBP sources and total average arrival rate 0.205291.

To our knowledge, the concept of a WFB/BFB mechanism is a novel development. In this paper, we develop the mechanisms for a basic discrete-time multiplexing scenario where, in each time slot, there is at most 1 packet arrival from each source and at most 1 packet transmission. The concept, however, is applicable in the more general setting of bulk-arrival bulk-departure queueing systems as well as in systems with no queueing, such as some time-slotted multiple access systems. The present paper therefore represents an initial study of the mechanisms in a relatively simple setting. The paper begins by defining the basic multiplexing and blocking scenario and explaining how the WFB mechanism may be used on top of various multiplexing schemes. The WFB mechanism is then derived in the case of 2 MMBP sources. The fundamental problem of deriving a WFB mechanism for an arbitrary number of Bernoulli sources is formulated as a linear programming problem. An explicit solution for BFB in the case of 3 Bernoulli sources is also given. Numerical results are presented to demonstrate the buffer savings that may be realized with the mechanism.

Asymptotic estimates for blocking probabilities in a large multi-rate loss network.

A.Simonian, J.W.Roberts[1], F.Théberge[2], R.Mazumdar[3]

[1] France Télécom - CNET, 38-40 r. Général Leclerc, 92131 Issy-les-Moulineaux, FRANCE
[2] McGill University, Dept. of Elec. Eng., Montreal, P.Q., H3A 2A7, CANADA
[3] INRS-Télécom., Université du Québec, Ile des Soeurs, P.Q., H3E 1H6, CANADA

Abstract

In this paper, asymptotic estimates for the blocking probability of a call pertaining to a given route in a large multi-rate circuit-switched network are given. Concentrating on light load conditions, these estimates are essentially derived by using probability change techniques applied to the distribution of the number of occupied links. This uniform expression is numerically validated via simple examples.

Keywords: multi-rate loss networks, blocking probability, large deviations, asymptotic estimate.

1 Introduction

Multi-rate loss network models are of use in the connection acceptance phase of currently envisaged ATM based broadband networks. The utility of such models is that they are robust w.r.t exact holding time distributions due to the *insensitivity property* and correctly capture the phenomenon of bandwidth requirements through proper interpretation of rates. One important feature of such models in the ATM context is their *size.* The single link problem has received much attention and exact results based on recurrence relations were first obtained by Kaufman [8] which were then generalized to the network case by Dziong and Roberts [4]. Such approaches are limited by computational complexity in the case of large systems and thus alternative approaches with exploits *largeness* were first proposed by Gazdicki, Lambadaris and Mazumdar [7]. More recently a re-derivation of the results in [7] via saddle-point approach in the same spirit of [14] was done by Mitra and Morrison [9].

The network case is much more difficult due to dependencies in flows along links. One approach is via the use of fixed-point heuristics [3] but it is difficult to glean qualitative information from such an approach.

Asymptotic results have been obtained in the framework of the so-called "large" networks, where mean traffic intensities and link capacities together grow to infinity while keeping the network load constant. Low, critical and heavy load conditions are then distinguished, corresponding to the case when the average load on each link is smaller than, equal to and larger than the capacity of this link, respectively. In this framework, asymptotic dominant terms for the blocking rates on a single link have been derived successively for each load condition by invoking probabilistic methods based on the theory of large deviations [7]. Using saddle-point techniques again for the single link case, a uniform asymptotic estimate has been also obtained [9], providing the three different expressions associated with low, critical and heavy load conditions given in [7] as special cases. In the case of a multi-link network, a limit result has been derived [6] for the blocking rate B_r along route r, namely

$$B_r \to 1 - \prod_j (1 - b_j)^{A_{jr}} \tag{1.1}$$

with a product over all links crossed by route r and where A_{jr} is the number of circuits required by this route on link j. Asymptotic formula (1.1) indicates that in the limit of large networks, blocking on the different circuits of link j arise independently with probability b_j. The set of all $b_j \in [0,1]$ is determined as the solution of a non-linear system of inequalities with size the number of links on the considered route. A positive b_j can here be seen to correspond to a heavily loaded link, whereas $b_j = 0$ is associated with a low or critically loaded link. In the same context, the convergence of the stationary distribution of the number of calls on each route towards a normal distribution has also been studied for critically loaded links in [5], where estimates similar to (1.1) have been provided. Such estimates have been further specified in [13] for the single link case.

Limit result (1.1) has been shown to work most efficiently in heavy load conditions only, notably when using it for fixed point heuristics as in [3]. On the other hand, in the context of broadband telecommunication networks, the order of magnitude of blocking probabilities required for the definition of connection or data burst acceptance criteria can correspond to small blocking probabilities in the range $10^{-3} - 10^{-5}$, rather than $10^{-1} - 10^{-2}$. Such acceptance criteria, in particular, have been studied in [9] in the case of a single link. It is therefore necessary to derive other approximations of possibly small blocking rates on a network in other traffic conditions, thus making assertion $b_j = 0$ in (1.1) more precise. In this respect, the aim of the present contribution is to derive asymptotic expressions for the blocking B_r along a given route when assuming low and critical load conditions, respectively. Space constraints only allow us to present the results for the lightly loaded case in this paper. The extended version of this paper [15] discusses the other cases.As mentioned above, the critical case has been already envisaged in [5], but approximations have not been made explicit in terms of network parameters (except in the single link case). The estimates we provide here are analytic expressions in terms of these parameters enabling us to highlight dependencies and simplify computational procedures.

The rest of the paper is organized as follows. The derivation of asymptotic estimates for blocking probabilities is addressed in Section 3 in light load condition. Arguments from large deviations theory, namely the probability change argument and the local Gaussian approximation for the shifted distribution, are here applied to the distributions of variables K. Our development can be seen as an extension to networks of that of [7] performed for a single link. Results are derived once the existence of the solutions for non-linear systems is assumed. A sufficient condition ensuring this existence is made explicit in Section 4 via asymptotic estimates for the distributions of variables M. The estimate is numerically validated in Section 6 through typical examples.

2 Notation and preliminaries.

Consider a network with J links and offered calls transfered via R fixed routes. Calls on route $\rho \in \{1, ..., R\}$ occur as a Poisson process with rate ν_ρ and their mean holding time is taken equal to unity. Flows associated with each route are mutually independent. The total number of circuits on link $j \in \{1, ..., J\}$ is denoted by C_j. A call pertaining to route ρ requires $A_{j\rho}$ circuits on link j (with

$A_{j\rho} = 0$ if $j \notin \rho$). Without loss of generality, the greatest common divisor d_j of the numbers $A_{j\rho}$, $j \in \rho$, is set to 1 (if $d_j \neq 1$, C_j and $A_{j\rho}$ can be replaced by $[C_j/d_j]$ and $A_{j\rho}/d_j$, respectively). In the following, $\mathbf{A}$ denotes the matrix with elements $A_{j\rho}$, $1 \leq j \leq J$, $1 \leq \rho \leq R$. Let M_ρ and

$$K_j = \sum_{1 \leq \rho \leq R} A_{j\rho} M_\rho \tag{2.1}$$

represent the number of calls currently in progress on route ρ and the number of circuits currently busy on link j, respectively.

Consider the probability distribution p (resp. q) of vector $\mathbf{M} = (M_1, ..., M_R)$ (resp. $\mathbf{K} = (K_1, ..., K_J)$) depending on all link capacities C_j, and the limit distribution P (resp. Q) of p (resp. q) corresponding to $C_j = +\infty$ for all j. In the latter case of infinite capacities, all M_ρ are mutually independent and have a Poisson distribution defined by

$$P(\mathbf{m}) = \prod_\rho e^{-\nu_\rho} \frac{\nu_\rho^{m_\rho}}{m_\rho !} \tag{2.2}$$

for $\mathbf{m} \in \mathbb{N}^R$. It is then known [6] that distribution p can be derived from P by truncation, that is,

$$p(\mathbf{m}) = P(\mathbf{m})/\mathcal{G} \tag{2.3}$$

for $\mathbf{0} \leq \mathbf{A}.\mathbf{m} \leq \mathbf{C}$, normalization constant $\mathcal{G}$ satisfying condition $\sum_{\mathbf{0} \leq \mathbf{A}.\mathbf{m} \leq \mathbf{C}} p(\mathbf{m}) = 1$. Equivalently, distribution q can be defined from Q by

$$q(\mathbf{k}) = Q(\mathbf{k})/\mathcal{G} \tag{2.4}$$

for $\mathbf{0} \leq \mathbf{k} \leq \mathbf{C}$, with identical normalization constant $\mathcal{G}$. From definition (2.1) and expression (2.2), the generating series $\Phi(\mathbf{z}) = \sum_{\mathbf{k} \geq \mathbf{0}} Q(\mathbf{k}) z_1^{k_1} ... z_J^{k_J}$ associated with distribution Q can be expressed as

$$\Phi(\mathbf{z}) = \exp\left[\sum_\rho \nu_\rho (z_1^{A_{1\rho}} ... z_J^{A_{J\rho}} - 1)\right]. \tag{2.5}$$

For each $l \in \{1, ..., J\}$, the notation L and $\mathbf{K}_L$ correspond in the following to a l-tuple $(j_1, ..., j_l)$, $1 \leq j_1 < ... < j_l \leq J$, and to the vector $(K_{j_1}, ..., K_{j_l})$, respectively (the specific sequence $(j_1, ..., j_l)$ associated with L is not made precise unless necessary and no index L is mentioned in the case when $l = J$). Distribution q_L (resp. Q_L) of vector $\mathbf{K}_L$ is defined by appropriate summation in (2.4). As in (2.5), the generating series $\Phi_L(\mathbf{z}_L)$ associated with distribution Q_L, $L = (j_1, ..., j_l)$, is given by

$$\Phi_L(\mathbf{z}_L) = \exp\left[\sum_\rho \nu_\rho (z_{j_1}^{A_{j_1\rho}} ... z_{j_l}^{A_{j_l\rho}} - 1)\right]. \tag{2.6}$$

For $l = 1$ and $L = j$, in particular, we derive by differentiation of $\Phi_j(z_j)$ at $z_j = 1$ that the expectation of distribution Q_j is $\sum_\rho \nu_\rho A_{j\rho}$. The latter is defined as the average load on link j. This link is said to be under low, critical or heavy load if its average load is smaller than, equal to or larger than its capacity C_j, respectively.

In the sequel, we multiply both mean traffic intensities and link capacities by a large parameter N so that the load of the network remains constant. For all ρ and j, the mean traffic intensity on route ρ and the capacity of link j are therefore $N\nu_\rho$ and NC_j, respectively, with $\nu_\rho = O(1)$, $C_j = O(1)$. In this so-called "large network" framework, distributions p, P and q, Q, q_L, Q_L consequently depend on N.

Let B_r denote the blocking probability of calls along a given route r. Given the state spaces $\mathcal{M} = \{\mathbf{m} \in \mathbb{N}^R \ / \ \forall j, \ \sum_\rho A_{j\rho} m_\rho \leq NC_j\}$ and $\mathcal{K} = \{\mathbf{k} \in \mathbb{N}^J \ / \ \forall j, \ K_j \leq NC_j\}$ for variables **M** and **K**, respectively, define the sets $\mathcal{M}_{jr} = \{\mathbf{m} \in \mathcal{M} \ / \ NC_j - A_{jr} < \sum_\rho A_{j\rho} m_\rho \leq NC_j\}$ and $\mathcal{K}_{jr} = \{\mathbf{k} \in \mathcal{K} \ / \ NC_j - A_{jr} < K_j \leq NC_j\}$ corresponding equivalently to blocking states for route r on link j. We have

$$B_r = p(\mathcal{M}_{1r} \cup ... \cup \mathcal{M}_{Jr}) = q(\mathcal{K}_{1r} \cup ... \cup \mathcal{K}_{Jr}).$$

Using Poincaré's formula for the probability of the union of events, we can write

$$B_r = \sum_{l=1}^{J} (-1)^{l-1} \beta_l \tag{2.7}$$

with

$$\beta_l = \sum_{1 \leq j_1 < ... < j_l \leq J} p(\mathcal{M}_{j_1,r} \cap ... \cap \mathcal{M}_{j_l,r}) = \sum_{1 \leq j_1 < ... < j_l \leq J} q(\mathcal{K}_{j_1,r} \cap ... \cap \mathcal{K}_{j_l,r}) \tag{2.8}$$

depending on r and parameter N.

3 Probability change method.

In this section, we address the estimate for large N of blocking probability B_r via the distributions q_L of variables $\mathbf{K}_L$ representing the number of occupied circuits on any l-tuple of links. For states corresponding to blocking events $\mathcal{K}_{j_1,r} \cap ... \cap \mathcal{K}_{j_l,r}$ introduced in the previous section, these distributions are approximated by means of tools from the theory of Large deviations [2], namely the probability change (or "shift") around the blocking events and the local approximation of shifted distributions by Gaussian densities.

3.1 Shifted distribution.

For given $l \in \{1, ..., J\}$, let L be the l-tuple $(j_1, ..., j_l)$. Replacing ν_ρ by $N\nu_\rho$ in expression (2.6), it is noted that the generating series of distribution Q_L is now $\Phi_L^N(\mathbf{z}_L)$. Let $\mathcal{W}_L$ denote some subset of $\mathbb{R}^l$ independent of N. As stated in [2, p.21], a logarithmic estimate for the probability $Q_L(N\mathcal{W}_L)$ that $\mathbf{K}_L/N \in \mathcal{W}_L$ can be asserted as

$$\lim_{N \uparrow +\infty} \frac{1}{N} \log Q_L(N\mathcal{W}_L) = - \inf_{\mathbf{w}_L \in \mathcal{W}_L} \mathcal{J}_L(\mathbf{w}_L)$$

with

$$\mathcal{J}_L(\mathbf{w}_L) = - \inf_{\mathbf{y}_L \in \mathbb{R}^l} [\log \Phi_L(e^{-y_{j_1}}, ..., e^{-y_{j_l}}) + \mathbf{w}_L . \mathbf{y}_L]$$

where $\mathbf{w}_L.\mathbf{y}_L$ stands for the scalar product of vectors $\mathbf{w}_L$ and $\mathbf{y}_L$ in $\mathbb{R}^l$. As detailed below, the exact (not only logarithmic) asymptotic estimate for $Q_L(N\mathcal{W}_L)$ where $\mathcal{W}_L$ is some neighborhood of capacity vector $\mathbf{C}_L$ is crucial for evaluating blocking probabilities. To determine such an exact estimate, some additional assumptions and properties enabling the use of a probability change are required. Let $I_L = \mathcal{J}_L(\mathbf{C}_L)$ as defined above for $\mathbf{w}_L = \mathbf{C}_L$. We first assume that this infimum I_L is attained at some point $\mathbf{y}_L = (y_{j_1}, ..., y_{j_l})$. In view of expression (2.6) for Φ_L and the definition of $\mathcal{J}_L$, such a point $\mathbf{y}_L$ must then satisfy the non-linear system

$$\sum_\rho \nu_\rho A_{j_s \rho} e^{-A_{j_1 \rho} \cdot y_{j_1}} ... e^{-A_{j_l \rho} \cdot y_{j_l}} = C_{j_s}, \quad s \in \{1, ..., l\} \tag{3.1}$$

and the corresponding infimum I_L can be written as

$$I_L = \sum_\rho \nu_\rho (1 - e^{-\mathbf{A}_{L\rho} \cdot \mathbf{y}_L}) - \mathbf{C}_L \cdot \mathbf{y}_L \tag{3.2}$$

where $\mathbf{A}_{L\rho}$ is the vector with components $A_{j\rho}$, $j \in L$. The existence of a point $\mathbf{y}_L$ with components satisfying system (3.1) can be ensured provided the Markov process representing the number of active calls in the network has an irreducible state space. Such a sufficient condition is discussed in Section 4.

For simplicity of index notation, we now proceed with the case when $l = J$ with corresponding solution to (3.1), $\mathbf{y} = (y_1, ..., y_J)$ (as detailed below, the case of any smaller dimension l can easily be deduced from the latter). Introduce then the "shifted" probability distribution $\overline{Q}$ (depending on $\mathbf{y}$) defined by

$$\overline{Q}(\mathbf{k}) = Q(\mathbf{k}) \frac{e^{-k_1 y_1} ... e^{-k_J y_J}}{\Phi^N(e^{-y_1}, ..., e^{-y_J})} \tag{3.3}$$

for all $\mathbf{k} \in \mathbb{N}^J$, with $\Phi(\mathbf{z})$ given in (2.5). A local central limit theorem for distribution $\overline{Q}$ can be stated as follows.

Lemma 3.1 *Given a solution* $\mathbf{y}$ *of system (3.1), let* Γ *be the matrix with elements*

$$\Gamma^{jk} = \sum_\rho \nu_\rho A_{j\rho} A_{k\rho} e^{-A_{1\rho} y_1} ... e^{-A_{J\rho} y_J}, \tag{3.4}$$

$1 \leq j, k \leq J$. *Denoting by* $m \leq J$ *the rank of* Γ, *distribution* $\overline{Q}$ *satisfies*

$$\overline{Q}(\mathbf{k}) = \frac{1}{N^{m/2}} \psi\left(\frac{\mathbf{k} - N\mathbf{C}}{\sqrt{N}}\right) + o\left(\frac{1}{N^{m/2}}\right)$$

for large N *and uniformly with respect to* $\mathbf{k}$; *function* $\psi : \mathbb{R}^J \to \mathbb{R}$ *is a Gaussian probability density independent of* N *with*

$$\psi(\mathbf{0}) = \frac{1}{\sqrt{(2\pi)^m D}}$$

where D *denotes the product of all non-zero eigenvalues of matrix* Γ.

Replacing in (3.3) Φ by Φ_L defined in (2.6), an asymptotic estimate similar to that of Lemma 3.1 can be obtained for the shifted distribution $\overline{Q}_L$ associated with vector $\mathbf{K}_L$, depending on a solution $\mathbf{y}_L$ of system (3.1). Matrix Γ in (3.4) is then replaced by the corresponding matrix Γ_L with size $l \times l$ given by

$$\Gamma_L^{jk} = \sum_\rho \nu_\rho A_{j\rho} A_{k\rho} e^{-A_{j_1\rho} y_{j_1}} ... e^{-A_{j_l\rho} y_{j_l}} \tag{3.5}$$

for $j, k \in L$ and density ψ is changed to corresponding density $\psi_L : \mathbb{R}^l \to \mathbb{R}$. For $l = 1$, in particular, corresponding Lemma 3.1 asserts that each distribution $\overline{Q}_j$ is asymptotically Gaussian with mean NC_j and variance $N\Gamma_j$ given by

$$\Gamma_j = \sum_\rho \nu_\rho A_{j\rho}^2 e^{-A_{j\rho} y_j} \tag{3.6}$$

where y_j is the solution of corresponding equation (3.1), that is,

$$\sum_\rho \nu_\rho A_{j\rho} e^{-A_{j\rho} y_j} = C_j. \tag{3.7}$$

Note that in this specific case, variance Γ_j is clearly non-zero and solution y_j is always uniquely defined. For order $l \geq 2$, one can provide the following simple assertion .

Lemma 3.2 *For any solution* $\mathbf{y}_L$ *to system (3.1), the rank* m_L *of corresponding covariance matrix* Γ_L *defined in (3.5) is equal to that of submatrix* $\mathbf{A}_L$ *with elements* $A_{j\rho}$, $j \in L, 1 \leq \rho \leq R$.

3.2 Main results.

Given the assumptions and lemmas stated in section 3.1, asymptotic estimates for blocking probabilities can now be asserted as follows. We state only the light load results for the sake of brevity. The so called critical load case which delineates (in terms of behavior) the light and heavy load (where the average load on each link of a route exceeds the capacity of the link) are given in [15]. It suffices to mention that the heavy load case gives the results in [6] as given in the introduction.

Proposition 3.1 *Assume that,*

- *for all* $L = (j_1, ..., j_l)$, $1 \leq l \leq J$, *any solution* $\mathbf{y}_L \in \mathbb{R}^l$ *of system (3.1) has negative components,*
- *and any* $2 \times R$ *submatrix of* $\mathbf{A}$ *has rank 2.*

We then have

$$B_r = \frac{1}{\sqrt{2\pi N}} \sum_{j, A_{jr} \neq 0} e^{-N.I_j} \frac{1}{\sqrt{\Gamma_j}} \frac{1 - e^{-y_j A_{jr}}}{1 - e^{-y_j}} \left[1 + O\left(\frac{1}{\sqrt{N}}\right)\right] \tag{3.8}$$

for large N, with

$$I_j = \sum_\rho \nu_\rho (1 - e^{-A_{j\rho} y_j}) - C_j y_j$$

and Γ_j *and* $y_j < 0$ *defined by (3.6) and (3.7), respectively.*

From (3.1), the negativity of components of solution $\mathbf{y}_L$ implies that $\sum_\rho \nu_\rho A_{j\rho} < C_j$ for all $j \in \{1, ..., J\}$, which corresponds to a low load regime for each link in the network.

Proof Writing (3.3) for the distribution Q_L associated with a solution $\mathbf{y}_L = (y_{j_1}, ..., y_{j_l})$ of (3.1), we have

$$Q_L(\mathbf{k}_L) = \Phi_L(e^{-y_{j_1}}, ..., e^{-y_{j_l}}) e^{k_{j_1} y_{j_1} + ... + k_{j_l} y_{j_l}} . \overline{Q}_L(\mathbf{k}_L)$$

where $\mathbf{k}_L = (k_{j_1}, ..., k_{j_l})$. Making $\Phi_L(e^{-y_{j_1}}, ..., e^{-y_{j_l}})$ explicit through (2.6), this equality also reads

$$Q_L(N\mathbf{C}_L - \mathbf{h}_L) = e^{-N.I_L} e^{-\mathbf{h}_L \cdot \mathbf{y}_L} . \overline{Q}_L(N\mathbf{C}_L - \mathbf{h}_L) \tag{3.9}$$

for fixed $\mathbf{h}_L$ and with I_L defined in (3.2). Considering $\mathbf{y}_L$ as an independent variable in expression (3.2) for $I_L = I_L(\mathbf{y}_L)$, we first have $I_L(\mathbf{0}_L) = 0$ and the MacLaurin formula asserts that there exists some $\theta \in]0, 1[$ such that $I_L(\mathbf{y}_L) = \mathbf{y}_L . \nabla I_L(\theta \mathbf{y}_L)$, where ∇ is the gradient operator with respect to $\mathbf{y}_L$. Equations (3.1) and the negativity of components of $\mathbf{y}_L$ then readily imply from the latter equality that $I_L > 0$.

Now, the normalizing constant $\mathcal{G}$ associated with distribution q in (2.4) can be written as

$$\mathcal{G} = \sum_{\mathbf{k} \leq N\mathbf{C}} Q(\mathbf{k}) = 1 - \sum_{\mathbf{h} \in \mathcal{N}_1 \cup ... \cup \mathcal{N}_J} Q(N\mathbf{C} - \mathbf{h}) \tag{3.10}$$

where $\mathcal{N}_j = \{\mathbf{h} / h_j < 0\}$. Using Poincaré's formula, the latter sum can be written as a sum of terms of the kind

$$\mathcal{G}_L = \sum_{\mathbf{h}_L \in \mathcal{N}_{j_1} \cap ... \cap \mathcal{N}_{j_l}} Q_L(N\mathbf{C}_L - \mathbf{h}_L)$$

for all $l \in \{1, ..., J\}$. From (3.9), each $Q_L(N\mathbf{C}_L - \mathbf{h}_L)$ is upper bounded by $e^{-N.I_L}e^{-\mathbf{h}_L.\mathbf{y}_L}$ and the summation $\sum \exp(-\mathbf{h}_L.\mathbf{y}_L)$ over $\mathbf{h}_L \in \mathcal{N}_{j_1} \cap ... \cap \mathcal{N}_{j_l}$ is finite since all components of $\mathbf{y}_L$ are negative. As a consequence, each $\mathcal{G}_L$ is of order $e^{-N.I_L}$ and thus tends to 0 exponentially fast since $I_L > 0$. This implies from (3.10) that $\mathcal{G}$ tends to 1. We consequently derive from (3.9) and the above discussion the estimate

$$q_L(N\mathbf{C}_L - \mathbf{h}_L) \sim e^{-N.I_L}e^{-\mathbf{h}_L.\mathbf{y}_L}.\overline{Q}_L(N\mathbf{C}_L - \mathbf{h}_L) \tag{3.11}$$

for distribution q_L with large N. Furthermore, applying Lemma 3.1 to distribution $\overline{Q}_L$ enables us to write

$$\overline{Q}_L(N\mathbf{C}_L - \mathbf{h}_L) = \frac{1}{N^{m_L/2}}\psi_L\left(\frac{\mathbf{h}_L}{\sqrt{N}}\right) + o\left(\frac{1}{N^{m_L/2}}\right) \tag{3.12}$$

where m_L is the rank of matrix Γ_L.

Condition $\mathbf{k}_L \in \mathcal{K}_{j_1,r} \cap ... \cap \mathcal{K}_{j_l,r}$ entails that $\mathbf{h}_L = N\mathbf{C}_L - \mathbf{k}_L$ is $O(1)$ since each component h_{j_s}, $s \in \{1, ..., l\}$, then lies in $\{0, 1, ..., A_{j_s r} - 1\}$. Density ψ_L being continuous at $\mathbf{0}_L$, the term $\psi_L(\mathbf{h}_L/\sqrt{N})$ in (3.12) thus tends to $\psi_L(\mathbf{0}_L)$ for increasing N and $\mathbf{k}_L \in \mathcal{K}_{j_1,r} \cap ... \cap \mathcal{K}_{j_l,r}$. From (3.11)-(3.12), each term β_l in (2.8) is then asymptotic to

$$\beta_l \sim \sum_L e^{-N.I_L}\frac{\psi_L(\mathbf{0}_L)}{N^{m_L/2}}\sum_{\mathbf{h}_L} e^{-\mathbf{h}_L.\mathbf{y}_L} \tag{3.13}$$

where $\mathbf{h}_L$ in the latter sum is $O(1)$. Estimates (3.11)-(3.12) also imply that $\log q_L(N\mathbf{C}_L)/N$ is asymptotic to I_L. As a consequence, given $L \subset L'$, we have $q_L(N\mathbf{C}_L) \geq q_{L'}(N\mathbf{C}_{L'})$ and thus by taking logarithms, $I_L \leq I_{L'}$. Now, using the assumption $\text{rank}(\mathbf{A}_L) \geq 2$ for $l \geq 2$ together with Lemma 3.2, we also have $m_L \geq 2$ for all $l \geq 2$. We thus derive from (3.13) and the above discussion that the dominant term β_l in the sum (2.7) corresponds to $l = 1$, and consequently $B_r \sim \beta_1 = \sum_j q(\mathcal{K}_{jr})$. Estimate (3.13) applied to $l = 1$ with $\psi_j(0) = 1/\sqrt{2\pi\Gamma_j}$, together with definitions (3.6) and (3.7) then gives

$$B_r \sim \sum_{j, A_{jr} \neq 0}^{J} e^{-N.I_j}\frac{1}{\sqrt{2\pi N\Gamma_j}}\sum_{h_j=0}^{A_{jr}-1} e^{-h_j y_j}.$$

The sum with respect to h_j eventually simplifies to give (3.8). ∎

4 Existence questions.

In the earlier section, we introduced some existence and regularity assumptions in order to derive asymptotic formulae for the blocking rate on a given route, essentially the existence of solutions for systems (3.1) and a condition on the rank of traffic matrix $\mathbf{A}$. The non-positivity of the components for solution $\mathbf{y}_L$ of system (3.1) has been seen, in particular, to be directly related to the underloaded or critically loaded regime in the network. In this section, we make the sufficient condition for the existence (and possible unicity) of solution $\mathbf{y}_L$ more explicit through the use of the distribution p for the number of active calls in the network. We also examine some generic situations which ensure the regularity of matrix $\mathbf{A}$.

Consider formula (2.8) where each β_l is expressed in terms of distribution p, that is, according to (2.3),

$$\beta_l = \frac{1}{\mathcal{G}} \sum_{1 \leq j_1 < ... < j_l \leq J} P(\mathcal{M}_{j_1,r} \cap ... \cap \mathcal{M}_{j_l,r}) \tag{4.1}$$

where P is the Poisson distribution defined in (2.2). Assume that the dominant value in the sum corresponds to terms $P(\mathbf{m})$ such that $\mathbf{m} = N\mathbf{x}$ with $\mathbf{x} = O(1)$ for large N. In the following, the non negativity of real components of vector $\mathbf{x} = (x_1, ..., x_R)$ is simply denoted by $\mathbf{x} \geq \mathbf{0}$. Applying Stirling's estimate $n! \sim n^n e^{-n}\sqrt{2\pi n}$ to (2.2) then entails that

$$P(\mathbf{m}) \sim e^{-NF(\mathbf{x})} \prod_{\rho/x_\rho>0} \frac{1}{\sqrt{2\pi N x_\rho}} \tag{4.2}$$

for large N, where

$$F(\mathbf{x}) = \sum_\rho \nu_\rho f\left(\frac{x_\rho}{\nu_\rho}\right)$$

with $f(u) = u \log u - u + 1$. Following arguments from the Laplace method for the estimate of sums in terms of a large parameter [10], the dominant term for $P(\mathcal{M}_{j_1,r} \cap ... \cap \mathcal{M}_{j_l,r})$ in (4.1) corresponds to the minimum of the exponential term in (4.2) under the constraint $\mathbf{m} \in \mathcal{M}_{j_1,r} \cap ... \cap \mathcal{M}_{j_l,r}$. As defined in Section 2, recall that $\mathcal{M}_{jr} = \{\mathbf{m} \in \mathcal{M},\ NC_j - A_{jr} < \sum_\rho A_{j\rho} m_\rho \leq NC_j\}$ for given j. As each A_{jr} remains fixed, the intersection $\mathcal{M}_{j_1,r} \cap ... \cap \mathcal{M}_{j_l,r}$ of considered constraints tends for increasing N to the set

$$X_L = \bigcap_{s=1}^{l} \{\mathbf{x} \geq \mathbf{0},\ \sum_\rho A_{j_s\rho} x_\rho = C_{j_s}\} \tag{4.3}$$

which is non-empty, provided the Markov process describing the number of active calls in the network reaches the state $(NC_{j_1}, ..., NC_{j_l})$. Such a condition is fulfilled if the state space of this process is irreducible. A detailed discussion as well as pathological cases are given in [15].

Now, for any l-tuple $L = (j_1, ..., j_l)$, $l > 1$, the minimization of the above defined function F over $X_L \neq \emptyset$ enables us to state conditions for the existence of solution $\mathbf{y}_L$ of system (3.1). These are formulated in the following assertion.

Proposition 4.1 *Provided the set X_L defined in (4.3) is non-empty, then non-linear system (3.1) has at least one solution. Moreover, if matrix $\mathbf{A}_L = (A_{j\rho})_{j\in L, 1\leq\rho\leq R}$, has maximal rank l, the solution of (3.1) is unique.*

The latter condition $\text{rank}(\mathbf{A}_L) = l$ necessarily entails that $l \leq R$. Note that for operational networks, there are usually more routes than circuits and condition $J \leq R$ is practically fulfilled. It also follows from Propositon 4.1 that a sufficient condition for system (3.1) to have a unique solution for any l-tuple L is also that $\text{rank}(\mathbf{A}) = J$, where traffic matrix $\mathbf{A}$ has full size $J \times R$. The same condition has been provided in [6] for the existence and unicity of quantities b_j in (1.1). The existence problem was here related to the minimization of function F with the constraint $\mathbf{A}.\mathbf{x} \leq \mathbf{C}$ (the constraint set is, in particular, never empty).

Assume now that the rank of matrix A_L is less than l for some l-tuple L. Then there exist scalars α_j, $j \in L$, such that $\sum_{j\in L} \alpha_j \mathbf{A}_j = \mathbf{0}$, where $\mathbf{A}_j$ denotes the jth row of matrix $\mathbf{A}$. If the set X_L of constraints associated with all indices $j \in L$ is non-empty, the necessary condition $\sum_{j\in L} \alpha_j C_j = 0$ is then derived. Such a specific condition is generally not fulfilled and we can therefore conclude as follows.

Proposition 4.2 *In the generic case and for $R \geq l$, the rank of matrix $(\mathbf{A}_L)$ is equal to l.*

Given the irreducibility of the system state space and condition $R \geq J$, the solution of (3.1) is thus generally unique as claimed in Prop.4.1. This ensures, in particular, that condition $\text{rank}(\mathbf{A}_L) = 2$ for $l = 2$ used in Prop. 3.1 and 3.2 holds true in the generic case. Given $L = (j, k)$, in particular, it is easily seen that $\text{rank}(\mathbf{A}_{(j,k)}) = 2$ provided there exist at least two routes not going through j and k simultaneously.

5 Numerical results.

In order to verify the accuracy of the results previously obtained, we conducted a wide range of numerical tests. We here validate estimate (3.8). We were mostly interested in analyzing the precision of our results when loading approaches the critical level, with blocking probabilities in the range 10^{-2} - 10^{-5}. All related tables and figures can be found at the end of the paper.

Space considerations allow us to present only one example depicted in figure 1 with 9 nodes and 13 links. Estimation (3.8) is compared to simulations with 95% confidence interval and Knapsack fixed-point heuristic. Case $N = 25$ is presented in Table 1 with blocking probabilities in the range $10^{-2} - 10^{-4}$, while case $N = 50$ is given in Table 4 with blocking probabilities of order $10^{-4} - 10^{-6}$. From these results, we observe that our estimates give good results for moderate blocking probabilities and are very precise for small blocking probabilities, although simulation results are expected to be less accurate in this situation.

ACKNOWLEDGMENT. This research was supported in part by a grant from the France-Québec Cooperative Research Programme.

References

[1] D.P.Bertsekas (1982), *Constrained optimization and Lagrange multiplier methods*, Academic Press

[2] J.A.Bucklew (1990), *Large deviations techniques in decision, simulation and estimate*, John Wiley

[3] S.Chung, K.W.Ross (1993), *Reduced load approximations for multirate loss networks*, IEEE Trans. Comm., 41, No.8, pp.1222-1231

[4] Z.Dziong, J.W.Roberts (1987), *Congestion probabilities in a circuit switched integrated services network*, Performance Evaluation, Vol.7, N^o 3

[5] P.J.Hunt, F.P.Kelly (1989), *On critically loaded loss networks*, Adv.Appl.Prob., 21, pp.831-841

[6] F.P.Kelly (1986), *Blocking probabilities in large circuit-switched networks*, Adv.Appl.Prob., 18, pp.473-505

[7] P.Gazdzicki, I.Lambadaris, R.R.Mazumdar (1993), *Blocking probabilities for large multirate Erlang loss systems*, Adv.Appl.Prob. 25, pp. 997-1009

[8] J. Kaufman (1981), *Blocking in a shared resources environment*, IEEE Trans. on Comm. **29**, 1474-1481

[9] D.Mitra, J.Morrison (1994), *Erlang capacity and uniform approximations for shared unbuffered resources*, IEEE/ACM Transactions on Networking, vol.2, N.6, pp.558-570

[10] J.D.Murray (1984), *Asymptotic analysis*, Springer Verlag

[11] V.V.Petrov (1975), *Sums of independent variables*, Springer Verlag

[12] C.R.Rao (1973), *Linear statistical inference and its applications*, John Wiley

[13] M.I.Reiman (1991), *A critically loaded multiclass Erlang loss system*, Queueing Systems, 9, pp.65-82

[14] A.Simonian (1992), *Analyse asymptotique des taux de blocage pour un trafic multidébit*, Annales des Télécommunications, 47, N^o 1-2, pp.56-63

[15] A. Simonian, J. Roberts, F. Théberge and R. Mazumdar, *Asymptotic estimates for blocking in large loss networks*, submitted for publication, June 1995.

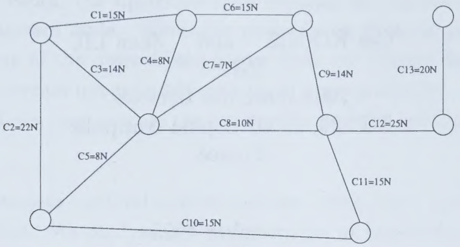

C1=15N
C6=15N
C3=14N
C4=8N
C7=7N
C9=14N
C13=20N
C2=22N
C8=10N
C12=25N
C5=8N
C11=15N
C10=15N

Figure 1: Network example

A_r	r	ν_r	Simulation	Knapsack	Est. (3.8)
1	1,2	4	(7.0-8.0)e-5	7.0e-5	7.23e-5
2	5,8,12	3	(4.0-4.3)e-4	4.4e-4	4.50e-4
3	6,9,12,13	2	(5.1-5.4)e-4	5.5e-4	5.57e-4
2	10,11,12,13	3	(3.5-3.7)e-4	3.9e-4	3.89e-4
1	3,4,6	6	< 1e-5	< 1e-5	2.01e-6
1	2,3,7,9	4	(7.0-8.0)e-5	7.0e-5	7.23e-5
2	8,11	1	(8.0-9.0)e-5	1.0e-4	1.11e-4
4	11,12,13	1	(7.3-7.7)e-4	8.1e-4	8.17e-4
2	1,2,10	3	(1.5-1.6)e-4	1.5e-4	1.54e-4
5	2	1	(4.0-4.3)e-4	4.1e-4	4.12e-4

Table 1: Blocking in Large Network of Figure 1, N = 50

On the Control of Tandems of Queues with Finite Buffers

Ger KOOLE and Zhen LIU
INRIA
2004 route des Lucioles
B.P. 93, 06902 Sophia Antipolis
France

September 1995

Abstract

Feedback congestion control policies are investigated for a model of an ATM network. We are concerned with a particular type of hop-by-hop control policies where the service at the upstream server is controlled in order to minimize the total loss rate in the two-stage system. We thus analyze a tandem queueing system with finite buffer capacities where both queues are fed by exogenous arrivals. The first server can be stopped so as to avoid congestion at the downstream queue. Assuming Markov modulated arrival processes for exogenous arrivals, we show, using dynamic programming techniques, that the optimal control policy for the minimization of the total loss rate (in both queues) has a monotone switching curve as a function of occupancies of the two buffers. The switching curve may vary as a function of the Markov environments of the arrival processes. Inspired by these theoretical results, we propose some simple approximate solutions for the congestion control. We provide numerical results comparing the loss rates with the optimal policies and the approximate solutions as well as some heuristics proposed in the literature. We further use simulation to investigate the robustness of the approximate solutions for the case of nonnegligible propagation delay of feedback information.

1 Introduction

One of the critical issues to be resolved for high-speed networks is the congestion control. Various feedback congestion control policies were proposed and analyzed in the literature, see e.g. [3, 5, 6, 7, 9]. One of the most studied issues is concerned with hop-by-hop control policies. For two adjacent nodes in a network, the upstream node controls the service rate in accordance with the state of the downstream node. When the propagation delay is significant, it is useful to predict the current state of the downstream queue from the delayed feedback information and departures from the upstream queue in between these time epochs (cf. e.g. [7]). However, most of proposed control policies does not use the current state of the upstream queue in the decision making.

In this paper we investigate optimal control policies using both upstream and downstream queue length information. We analyze a particular type of hop-by-hop control policies where the service at the upstream server is controlled in order to minimize the total loss rate in the two-stage system. We thus consider a tandem queueing system with finite buffer capacities where both queues are fed by exogenous arrivals. The first server can be stopped so as to avoid congestion at the downstream queue.

Assuming Markov modulated arrival processes for exogenous arrivals, we show, using dynamic programming techniques, that the optimal control policy for the minimization of the total loss rate (in both queues) has a monotone switching curve as a function of occupancies of the two buffers. The switching curve may vary as a function of the Markov environments of the arrival processes. Inspired by these theoretical results, we propose some simple approximate solutions for the congestion control. We provide numerical results comparing the loss rates with the optimal policies and the approximate solutions as well as some heuristics proposed in the literature.

We further use simulation to investigate the robustness of the approximate solutions for the case of nonnegligible propagation delay of feedback information.

2 Tandem Queueing System Model

In this paper we consider a tandem queueing system (see Figure 1) with finite-capacity buffers B_1 and B_2 and single servers of capacity c_1 and c_2. There are two streams of exogenous arrivals of customers in the system, one for queue 1 and the other for queue 2. Customers finishing service at queue 1 enter queue 2 (provided there is room in the buffer 2) for service. Customers finishing service at queue 2 leave the system. Both queues apply FIFO (first in first out) service discipline. A customer is lost when it finds the buffer full at arrival.

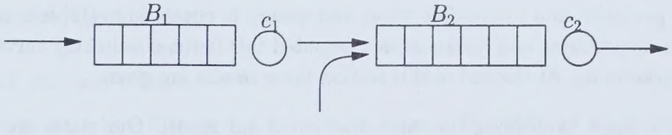


Figure 1: Tandem queueing model.

We will consider two models with Markov arrival processes: discrete time and continuous time models. In the discrete time case, the numbers of exogenous arrivals at each time slot are modeled by two stochastic sequences $\{A_{1,n}(Y_n)\}_n$ and $\{A_{2,n}(Y_n)\}_n$, where $\{Y_{j,n}\}_n$ is a finite-state, irreducible, aperiodic and homogeneous Markov chain on the set $\mathcal{S} := \{1, 2, \ldots, K\}$, and, for $j = 1, 2$ and $1 \leq i \leq K$, $\{A_{j,n}(i)\}_n$ are $2K$ mutually independent renewal sequences of $I\!N_+$-valued random variables (r.v.'s), further independent of the Markov chain $\{Y_n\}_n$. The service times are unit for all customers. The server capacities c_1 and c_2 represent the maximum numbers of customers that can be served at each time slot. Let $\beta_{jk}(i) = P[A_{j,n}(i) = k]$ be the probability of having k exogenous arrivals at queue j, $j = 1, 2$, within one time slot given the Markov chain Y_n is in state $i \in \mathcal{S}$. Let λ_{uv} be the transition probability of the Markov chain Y_n from state $u \in \mathcal{S}$ to $v \in \mathcal{S}$.

In the continuous time model, customers arrive from outside the system to queue j, $j = 1, 2$, according to a Markov Modulated Poisson Process (see [4] and references therein) with arrival rate $r_j(Z(t))$ at time t, where $(Z(t), t \geq 0)$ is an irreducible Markov process on $\mathcal{S} = \{1, 2, \ldots, K\}$ with transition rates λ_{uv} for $u, v \in \mathcal{S}$. The service times at both queues are i.i.d. exponentially distributed with rates c_1 and c_2 for queue 1 and queue 2.

For both models, there is a controller at queue 1 who decides whether or not to start the service of a new customer in order to minimize the total number of losses in the system. The decision epochs are customer arrival times and departure times at queue 1. There is no control at the second queue. But it is clear that the optimal control of the second queue is always to service. The controller has available to it the state of the Markov chain and the feedback information of queue 2: its queue length. We shall consider two cases concerning the availability of this feedback information: instantaneous feedback information and delayed feedback information.

3 Optimal Control without Propagation Delay

In this section we investigate the structure of the optimal policy for the case without propagation delay. This means that the first server knows not only the state of its own queue, but also the current state of the other queue. In addition, we assume that the controller has knowledge on the state of the Markov chain which drives the arrival streams.

This makes our control problem a Markov decision with full information. We analyze it using a dynamic programming approach. Using recently developed techniques (in [1]) we deal with the continuous and the discrete time models at the same time. For the continuous time model this builds heavily on uniformization (see [8]). For a general reference on dynamic programming see Bertsekas [2].

The result states that there is a switching curve which is increasing in the queue lengths. This result can be seen as a generalization of the work in Weber & Stidham [10] to finite buffers. Work on the generalization to multiple queue and queues in circles as in [10] is currently going on. For different problems and instances we computed this optimal switching curve, still using dynamic programming. At the end of this section these results are given.

We start the analysis by defining the value-function of our model. Our states are of the form (v, x), with $v \in \mathcal{S}$ the state of the environment and $x = (x_1, x_2) \in \{0, \ldots, B_1\} \times \{0, \ldots, B_2\}$ the queue lengths. The value function V_n is build from four different operators, each representing a possible event in the system.

The operators T_{A_1} and T_{A_2} are concerned with arrivals to the queues. Note that the cost is 1 if the corresponding buffer is full.

$$T_{A_1}f(v,x) = \begin{cases} f(v,x_1+1,x_2) & \text{if } x_1 < B_1 \\ 1+f(v,x) & \text{otherwise} \end{cases}$$

$$T_{A_2}f(v,x) = \begin{cases} f(v,x_1,x_2+1) & \text{if } x_2 < B_2 \\ 1+f(v,x) & \text{otherwise} \end{cases}$$

Operator T_{D_1} represents the server at the first queue. Here there is a choice between serving and not serving, explaining the minimization. For the moment we assume that no service takes place if the second queue is full. Later on we will see that this is indeed optimal.

$$T_{D_1}f(v,x) = \begin{cases} f(v,x) & \text{if } x_1 = 0 \text{ or } x_2 = B_2 \\ \min\{f(v,x_1-1,x_2+1), f(v,x)\} & \text{otherwise} \end{cases}$$

Finally T_{D_2} is concerned with departures from the second queue. These are uncontrolled. We will see that even if they were controlled, then it is optimal to control at full rate.

$$T_{D_2}f(v,x) = \begin{cases} f(v,x_1,x_2-1) & \text{if } x_2 > 0 \\ f(v,x) & \text{otherwise} \end{cases}$$

With these dynamic programming operators we can define the value function. For the discrete time model we would get

$$V_{n+1}(v,x) = \sum_w \lambda_{vw} T_{D_2}^{c_2} T_{D_1}^{c_1} \sum_k \beta_{1k_1}(w) T_{A_1}^{k_1} \sum_k \beta_{2k_2}(w) T_{A_2}^{k_2} V_n(w,x),$$

where T^s denotes the s-fold convolution of the operator T. Thus successively the following events happen: the environment changes state, a maximum of c_2 customer leave queue 2 (depending on the number of customers available), at most c_1 customer go from queue 1 to queue 2, where for each customer the optimal decision is taken, which is equivalent to saying that the optimal batch size is determined. After that, customers enter the system at both queues. For those which are blocked costs are incurred.

For the continuous time model we would get, after uniformization (meaning that $\sum_w \lambda_{vw} + r_1(v) + r_2(v) + c_1 + c_2 = 1$ for all v),

$$V_{n+1}(v,x) = \sum \lambda_{vw} V_n(w,x) + r_1(v) T_{A_1} V_n(v,x) + r_2(v) T_{A_2} V_n(v,x) + \\ c_1 T_{D_1} V_n(v,x) + c_2 T_{D_2} V_n(v,x).$$

We always take $V_0(v,x) = 0$.

Our goal is to show the following.

Theorem 3.1 *For each v and n*

$$V_n(v,x_1,x_2+1) - V_n(v,x_1+1,x_2) \le V_n(v,x_1,x_2+2) - V_n(v,x_1+1,x_2+1)$$

and

$$V_n(v,x_1+1,x_2+1) - V_n(v,x_1+2,x_2) \le V_n(v,x_1,x_2+1) - V_n(v,x_1+1,x_2),$$

and there is an increasing switching curve below which it is optimal to serve queue 1.

Proof. In the lemma below both inequalities are proven. Here we just show that from this we can conclude that there is an increasing switching curve. If $V_n(v,x_1,x_2+1)-V_n(v,x_1+1,x_2)\le 0$ (≥ 0), then it is optimal (not) to serve at queue 1, as the terms occurring in the minimization of T_{D_1} are of this form. The first inequality states that $V_n(v,x_1,x_2+1)-V_n(v,x_1+1,x_2)$ is increasing in x_2, thus for fixed v and x_1 there is a level x_2^* such that it is optimal for values of x_2 to serve below, but not above this level. Similarly it follows from the second inequality that for fixed v and x_2 that there is a level x_1^* such that it is optimal for values of x_1 to serve only above that level. This gives the form of the switching curve. □

Lemma 3.1 *For each v and n and all possible values of x the following 7 inequalities hold:*

$$V_n(v,x_1,x_2+1)+V_n(v,x_1+1,x_2+1)\le V_n(v,x_1+1,x_2)+V_n(v,x_1,x_2+2) \tag{1}$$

$$V_n(v,x_1+1,x_2)+V_n(v,x_1+1,x_2+1)\le V_n(v,x_1,x_2+1)+V_n(v,x_1+2,x_2) \tag{2}$$

$$V_n(v,x_1+1,x_2)+V_n(v,x_1,x_2+1)\le V_n(v,x_1,x_2)+V_n(v,x_1+1,x_2+1) \tag{3}$$

$$V_n(v,x_1+1,x_2)\le 1+V_n(v,x_1,x_2) \tag{4}$$

$$V_n(v,x_1,x_2+1)\le 1+V_n(v,x_1,x_2) \tag{5}$$

$$V_n(v,x_1,x_2)\le V_n(v,x_1+1,x_2) \tag{6}$$

$$V_n(v,x_1,x_2)\le V_n(v,x_1,x_2+1) \tag{7}$$

The meaning of (1) and (2) is given in the proof of theorem 3.1. The other inequalities are needed to prove (1) and (2). Equation (3) is known as supermodularity. These three inequalities would also result if the results from [10] were applied to our tandem system. The last four inequalities are needed to deal with the boundary. Equations (4) and (5) say that it is better to have a customer extra in queue than a loss. Equations (6) and (7) state that it is best to have as less customers as possible.

Proof. We use induction to n. Write f for $V_n(v,\cdot)$, for some arbitrary v. Assuming that f satisfies all 7 inequalities, we show that $T_{A_1}f$, $T_{A_2}f$, $T_{D_1}f$ and $T_{D_2}f$ do as well. This completes the induction step, as the inequalities are closed under linear combinations or concatenations, and because v is arbitrary.

Note that combination of (1) and (3) gives

$$2f(x_1,x_2+1)\le f(x_1,x_2)+f(x_1,x_2+2), \tag{8}$$

and that combination of (2) and (3) gives

$$2f(x_1+1,x_2)\le f(x_1,x_2)+f(x_1+2,x_2), \tag{9}$$

thus we have convexity in both components, for all possible values of x.

Propagating (4)–(7) is straightforward, therefore we will only concern us with (1)–(3). We consider all 4 dynamic programming operators, and indicate the non-trivial steps, meaning that we need more than jsut induction.

- T_{A_1}. (1): if $x_1 + 1 = B_1$, we get (8). (2): if $x_1 + 2 = B_1$, we get (4). (3): trivial.
- T_{A_2}. (1): if $x_2 + 2 = B_2$, we get (5). (2): if $x_2 + 1 = B_2$, we get (9). (3): trivial.
- T_{D_1}. For each inequality, denote with a_1 (a_2) the optimal action in the first (second) term of the right hand side. We do not have to care about $x_1 = 0$ or $x_2 = B_2$, in these we simply take a_1 or $a_2 = 0$. (1): easy if $a_1 = a_2$ (note that if $a_1 = a_2 = 1$ that these actions are also allowed in both terms at the l.h.s.), $a_1 = 0$, $a_2 = 1$ does not occur due to the structure of the optimal policy, $a_1 = 1$, $a_2 = 0$ gives for properly chosen actions at the l.h.s. an equality. (2): $a_1 = 0$, $a_2 = 1$ is the only case to be considered, results in an equality. (3): $a_1 = 1$, $a_2 = 0$ gives (9), $a_1 = 0$, $a_2 = 1$ gives (8).
- T_{D_2}. (1): if $x_2 = 0$, we get (7). (2): if $x_2 = 0$, we get (9). (3): trivial. □

Remark. Our model could be generalized slightly by taking

$$T_{D_1} f(v, x) = \min\{1 + f(v, x_1 - 1, x_2), f(v, x)\}$$

if $x_2 = B_2$ and

$$T_{D_2} f(v, x) = \min\{f(v, x_1, x_2 - 1), f(v, x)\}$$

if $x_2 > 0$. However, this is equivalent to the previous definitions, as the added actions are suboptimal, by (4) and (7).

Thus having established that there is an optimal increasing switching curve, we compute these switching curves for 3 different models: a continuous time model with Poisson arrivals (the "Poisson" model), a continuous time model with for each queue an on-off source (the "on-off" model), and a discrete time model with a single state environment (the "discrete" model). The Poisson model represents smooth traffic; both with the on-off model and with the discrete model bursty traffic can be modeled.

Typical examples are given in the tables below, for $B_1 = 15$ and $B_2 = 10$. Note that the optimal policy is only given for $x_1 > 0$ and $x_2 < B_2$. A $*$ in a table means that both actions are optimal, numerically speaking.

x_2:9	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
8	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
7	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
6	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
x_1:	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5

Table 1. Continuous model, $r = (1.00, 1.00)$, $c = (1.25, 2.5)$

x_2:9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*
8	0	0	0	0	0	0	0	0	0	0	0	0	0	*	1
7	0	0	0	0	0	0	0	0	0	0	0	0	*	1	1
6	0	0	0	0	0	0	0	0	0	0	0	*	1	1	1
5	0	0	0	0	0	0	0	0	0	0	*	1	1	1	1
4	0	0	0	0	0	0	0	0	0	*	1	1	1	1	1
3	0	0	0	0	0	0	0	0	*	1	1	1	1	1	1
2	0	0	0	0	0	0	0	*	1	1	1	1	1	1	1
1	0	0	0	0	0	0	*	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
x_1:	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5

Table 2. Discrete model, $c_j = 1$, $\beta_{j0} = 0.9$, $\beta_{j4} = 0.1$, $j = 1, 2$

For the on-off model the optimal policy depends also on the state of the arrival process. This may be an undesirable feature, as this information might not be available. However, to find the optimal policy for the case without information on the arrival process we have to solve a control problem with partial observations, and this is practically impossible because of the size of the current model.

The rates given for the IPP's are the rates at which the environment changes from "on" to "off" and v.v. In one of the states customers are generated with rate r.

x_2:9	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
8	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
7	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
6	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
x_1:	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5

Table 3a. State IPP's (0,0)

x_2:9	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
8	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
x_1:	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5

Table 3b. State IPP's (1,0)

x_2:9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
6	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
5	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
4	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
3	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
x_1:	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5

Table 3c. State IPP's (0,1)

x_2:9	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
8	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
7	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
6	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
5	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
x_1:	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5

Table 3d. State IPP's (1,1)

Table 3. On-off model, rates IPP's=(0.10,0.10), (0.10,0.10), $r = (2.00, 2.00)$, $c = (1.25, 2.5)$

4 Comparison Between the Optimal Policy and Heuristics

Although these optimal policies are relative easy to compute, it would still be desirable to find policies which are close to optimal and which are simpler to describe. This would certainly play a role in models with more than two queues. Therefore we came up with different simple rules of thumb, for which we compared the performance with the optimal policy for different values of the parameters.

These rules of thumb were (formulated for the continuous model):

- Balanced policy: serve if $x_1 > x_2$.

- Proportionally balanced policy: serve if $x_1/B_1 > x_2/B_2$ or $x_2 = 0$.
- Complex policy: serve if $\frac{x_1 r_1}{(B_1 - x_1) r_1} > \frac{x_2 r_2}{(B_2 - x_2) c_2}$ or $x_2 = 0$.
- Threshold policy: serve if $x_2 < T$, with T the threshold. (This policy has already been studied in the literature, see the introduction.)
- Full service policy: serve always (i.e., when $x_1 > 0$).

Note that all these policies (even the full service policy) stop service if $x_2 = B_2$, that is, if queue 2 is full. Note also that if $B_1 = B_2$ several policies coincide. We actually computed the results for several more policies, and we considered several variants of the policies considered here (for example, we looked at the balanced policy with a shifted switching curve). To keep the paper concise, we prefer to restrict ourselves to the policies given above.

In the four tables below we compare, for the parameter values for which we gave the optimal policies above, the rules of thumb given here. The numbers given are the average numbers of blocked customers in the system.

Optimal	0.025993
Balanced	0.033986
Prop. balanced	0.041915
Complex	0.031763
Threshold $T = 5$	0.058380
Threshold $T = 8$ (optimal threshold value)	0.028227
Full service	0.036541

Table 4. Continuous model, $r = (1.00, 1.00)$, $c = (1.25, 2.5)$

Optimal	0.015912
Balanced	0.022758
Prop. balanced	0.018347
Threshold $T = 3$ (optimal threshold value)	0.019557
Threshold $T = 7$	0.033483

Table 5. Discrete model, $c_j = 1$, $\beta_{j0} = 0.9$, $\beta_{j4} = 0.1$, $j = 1, 2$

Optimal	0.006914
Balanced	0.008448
Prop. balanced	0.009219
Threshold $T = 5$	0.010058
Threshold $T = 7$ (optimal threshold value)	0.007900

Table 6. On-off model, rates IPP's=(0.10,0.10), (0.10,0.10), $c = (2.00, 2.00)$, $c = (1.25, 2.5)$

Overall, the balanced policy behaves very well. Note that the threshold policy for the optimal T does even better; it is not clear however how to find this level of T, otherwise than by computing the average loss for each value of T. The numbers show that it is difficult to find a good value of T without doing the computations.

5 Effect of Propagation Delays

Consider now the case where the feedback information on the downstream queue is delayed. Encouraged by the results of the previous section we study only the rules of thumb. Dynamic programming is not applicable for this case. The comparisons were thus performed by simulation with confidence interval 95%. We consider the on-off arrival models. For queue 1, the transition rates from "on" to "off" and from "off" to "on" are both equal to 0.1. For queue 2, the transition rate from "on" to "off" is 0.1 and from "off" to "on" is 0.5. Queue 1 has buffer capacity 20 and service capacity 1. Queue 2 has buffer capacity 10 and service capacity 2.

In Fig. 2 we present the total loss rates of Balanced, Proportional Balanced and Complex, as well as Full Service policies as functions of propagation delays of feedback information. The results reported in the figure were obtained by taking the arrival rates (at "on" state) 1 for queue 1 and 1.2 for queue 2.

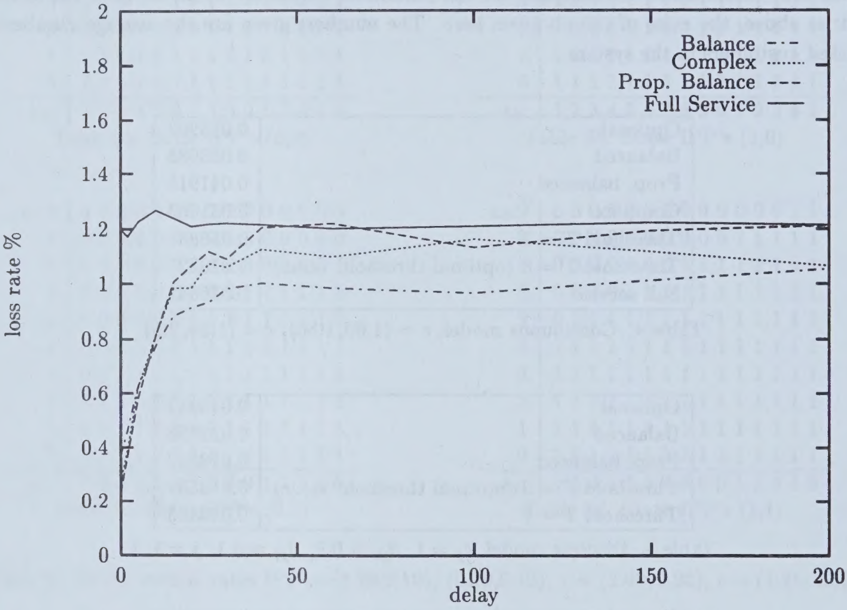


Figure 2: Effect of propagation delays

In Fig. 3 we present the total loss rates of the Balanced policy in comparison with the Full Service policy as functions of the workload at the downstream queue, where the arrival rate (at "on" state) of queue 1 is fixed to 1, and we vary the arrival rate at queue 2. Two curves are presented for the Balanced policy, one for the propagation delay 100 and other without propagation delay.

It can be seen from both figures that the advantage of the Balanced policy over Full Service policy diminishes as the propagation delay increases.

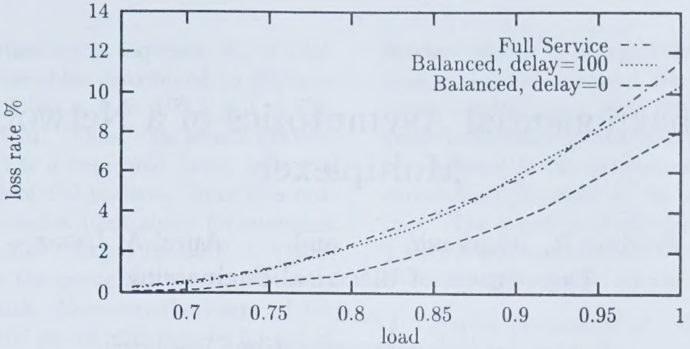


Figure 3: Effect of workload at the downstream queue

References

[1] E. Altman and G.M. Koole, "On Submodular Value Functions of Dynamic Programming", Working paper, 1995.

[2] D.P. Bertsekas, "Dynamic Programming", Prentice-Hall, 1987.

[3] K. W. Fendick, M. A. Rodrigues, A. Weiss, "Analysis of a Rate-Based Feedback Control Strategy for Long Haul Data Transport", *Perf. Eval.*, **16** (1992), pp. 67–84.

[4] W. Fischer and K. Meier-Hellstern, "The Markov-Modulated Poisson Process (MMPP) Cookbook", *Perf. Evaluation*, **18**, pp. 149-172, 1992.

[5] P. P. Mishra, H. Kanakia, "A Hop by Hop Rate-Based Congestion Control Scheme", *SIGCOMM 92*, pp. 112-123, 1992.

[6] D. Mitra, "Asymptotically Optimal Design of Congestion Control for High Speed Data Networks", *IEEE Trans. on Communications*, **40** (1992), pp. 301–311.

[7] G. Ramamurthy, B. Sengupta, "A Predictive Hop-by-Hop Congestion Control Policy for High Speed Networks", *INFOCOM 93*, pp. 1033-1041, 1993.

[8] R.F. Serfozo, "An equivalence between continuous and discrete time Markov decision processes", Operations Research 27:616–620, 1979.

[9] Y. T. Wang, B. Sengupta, "Performance Analysis of a Feedback Congestion Control Policy Under Non-Negligible Propagation Delay", *SIGCOMM 91*, pp. 149-157, 1991.

[10] R.R. Weber and S. Stidham, Jr., "Optimal Control of Service Rates in Networks of Queues", Advances in Applied Probability 19: 202–218, 1987.

Subexponential Asymptotics of a Network Multiplexer

Predrag R. Jelenković and Aurel A. Lazar
Department of Electrical Engineering
and
Center for Telecommunications Research
Columbia University, New York, NY 10027-6699
e-mail: {aurel, predrag}@ctr.columbia.edu

Abstract

For a Markov-modulated random walk with negative drift and long-tailed right tail we prove that the ascending ladder height matrix distribution is asymptotically proportional to a long-tailed distribution. This result enable us to generalize a recent result on subexponential asymptotics of a Markov-modulated M/G/1 queue to subexponential asymptotics of a Markov-modulated G/G/1 queue.

For a class of processes constructed by embedding a Markov chain into a subexponential renewal process we prove that the autocorrelation function has a subexponential tail. Furthermore, we prove that when a fluid flow queue is fed by these processes the queue length distribution is asymptotically proportional to its autocorrelation function.

1 Introduction

Under the variety of assumptions of Cramér type (exponentially bounded marginals and autocorrelation function) many published results have shown that the queue length distribution of a network multiplexer has exponential asymptotics, i.e., $\mathbb{P}[\mathrm{Q} > \mathrm{x}] \sim \alpha e^{-\theta^* \mathrm{x}}$ as $\mathrm{x} \to \infty$. Some authors have argued that an even simpler approximation holds $\mathbb{P}[\mathrm{Q} > \mathrm{x}] \sim e^{-\theta^* \mathrm{x}}$. This has led to the development of the so called effective bandwidth based admission control (see [6, 10, 13, 12, 17]). However, this approach may often lead to poor approximations [8, 14], which may result in a significant underutilization of network resources. The shortcoming of the single exponential approximation often manifests itself with arrival processes that span over multiple time scales. This has been independently shown in [14, 18]. A recursive numerical algorithm for computing the queue length distribution with multiple time scale arrivals can be found in [16].

The main motivation for this work are the results presented in [11] which show that the (marginal) distribution function and the autocorrelation function of the arrival processes that appear in communications networks may have a long (subexponential) tail. For such processes the Cramér type conditions are not satisfied.

To get some feeling about the behavior of the queue when the arrival process has a long tailed distribution let us examine the following example. (All the examples in this paper are calculated using the z-transform technique and Mathematica 2.2.)

Example 1 Consider a discrete time queue with a service rate of one packet per slot ($c = 1$) and an arrival process characterized by a sequence A_n of i.i.d. random variables distributed as $\mathbb{P}[A_0 = 0] = 0.2, \mathbb{P}[A_0 = i] = d/i^6, 1 \leq i \leq 150$, $d = 0.77151$. Thus, this source (arrival process) has a truncated heavy tail with peak rate of 150 packets. Since this process is bounded from above its cumulant function $\mathbb{E}e^{\theta A_0}$ exists for all $\theta > 0$ and, therefore, the queue tail is asymptotically exponential. However, the range of the exponential asymptotic may be far out of the relevant range of probabilities. On the right-hand side of figure 1 we can see that the exponential asymptotics starts to work for very small probabilities (roughly smaller than 10^{-40}). However, in the relevant range of probabilities ($10^{-4} - 10^{-10}$) we see, on the left-hand side of figure 1, that the exponential approximation fails. In this region, queue length probabilities have a functional form approximately proportional to the integrated tail of A_n ($1/i^5$).

Thus, we can see that *even in the case of bounded heavy tailed arrivals*, for which the queue length asymptotics is eventually exponential, the relevant part of the queue length distribution may be subexponential.

In this paper we examine the queue length distribution when the Cramér type conditions are replaced by subexponential assumptions.

The paper is organized as follows. In section 2 we give the basic definitions and results on subexponential distributions. At the end of that section (subsection 2.2) the classical result on the subexponential $GI/GI/1$ queue asymptotics is presented in Theorem 1. This result has been recently generalized to the Markov-modulated $M/G/1$ queue [4]. Our further generalization of this result to the Markov-modulated $G/G/1$ queue is presented in section 3. In the rest of the paper (section 4) we consider a class of processes that have subexponentially correlated arrivals. This processes are obtained by embedding Markov chains in a stationary subexponential renewal process. When this processes are fed into a fluid flow queue the queue length distribution is *asymptotically proportional to the autocovariance (autocorrelation) function of the arrival process.* The paper is concluded in section 5 with a brief application of these results to broadband network admission control.

2 Subexponential Distributions

Definition 1 *A distribution function F with $F(0-) = 0$ is called* long-tailed *($F \in \mathcal{L}$) if*

$$\lim_{x\to\infty} \frac{1 - F(x-y)}{1 - F(x)} = 1, \quad y \in \mathbb{R}. \tag{1}$$

Definition 2 *A distribution function F with $F(0-) = 0$ is called* subexponential *($F \in \mathcal{S}$) if*

$$\lim_{x\to\infty} \frac{1 - F^{*2}(x)}{1 - F(x)} = 2, \tag{2}$$

where F^{*2} denotes the 2-nd convolution of F with itself.

The class of subexponential distributions was first introduced by Chistakov [7]. The definition is motivated by the simplification of the asymptotic analysis of the convolution tails. Some examples of distribution functions in $\mathcal{S}$ are:

(I) the Pareto family

$$F(x) = 1 - (x - \beta + 1)^{-\alpha},$$

$x > \beta > 0, \alpha > 0.$

(II) the lognormal distribution

$$F(x) = \Phi\left(\frac{\log x - \mu}{\sigma}\right), \quad \mu \in \mathbb{R}, \sigma > 0,$$

where Φ is the standard normal distribution.

(III) Weibull distribution

$$F(x) = 1 - e^{-\alpha x^{\beta}},$$

for $0 < \beta < 1, \alpha > 0.$

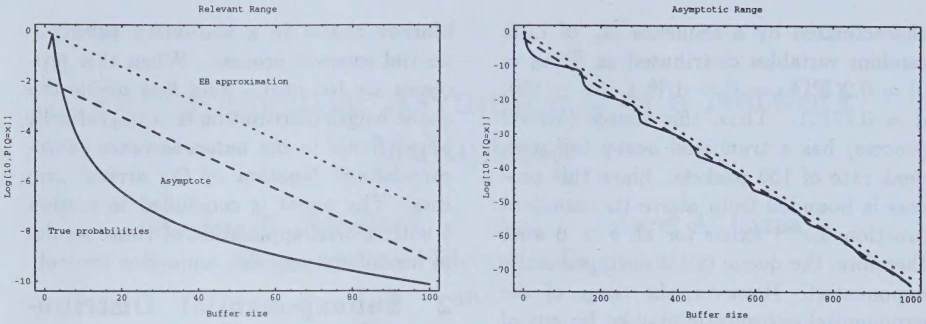


Figure 1: Illustration for Example 1.

(IV)

$$F(x) = e^{-x(\log x)^{-a}},$$

for $a > 0$.

(V) Benktander Type I distribution

$$F(x) = 1 - cx^{-a-1}x^{-b\log x}(a + 2b\log x),$$

$a > 0, b > 0$, and c appropriately chosen.

(V) Benktander Type II distribution

$$F(x) = 1 - cax^{-(1-b)}\exp\{-(a/b)x^b\},$$

$a > 0, 0 < b < 1$, and c appropriately chosen.

2.1 Basic Results

In what follows we will state an important result from the literature on subexponential distributions. The general relation between $\mathcal{S}$ and $\mathcal{L}$ is the following.

Lemma 1 *(Athrey and Ney, [5])* $\mathcal{S} \subset \mathcal{L}$.

Lemma 2 *If* $F \in \mathcal{L}$ *then* $(1 - F(x))e^{\alpha x} \to \infty$ *as* $x \to \infty$, *for all* $\alpha > 0$.

Note: Lemma 2 clearly shows that for long tailed distributions Cramér type conditions are not satisfied.

To simplify the notation let us denote by $\bar{F}(x) \overset{def}{=} 1 - F(x)$. The following lemma, used in the proof of theorem 4, is from [4].

Lemma 3 *Let* $G = \{G_{ij}\}$ *be a matrix of non-negative measures such that* $\|G\| \overset{def}{=} G(0, \infty)$ *is substochastic (the spectral radius is* < 1*). If there exists some probability distribution* $G \in \mathcal{S}$ *such that* $G_{ij}(x) \sim l_{ij}\bar{G}(x)$ *as* $x \to \infty$ *for some matrix* $L = \{l_{ij}\}, 0 < l_{ij} < \infty$, *then*

$$\sum_{n=0}^{\infty} \overline{G^{*n}}(x)(I - \|G\|) \sim (I - \|G\|)^{-1}L\bar{G}(x).$$

2.2 The GI/GI/1 queue with subexponential arrivals

Let $\{A_t, C_t, t \geq 0\}$ be a sequence of random variables (on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$). Then, for any initial random variable Q_0 the following (Lindley's) equation

$$Q_{t+1} = (Q_t + A_t - C_t)^+ \qquad (3)$$

defines the queue length process $\{Q_t\}$; throughout we deal with a stationary version of this process. The dynamics of a broadband network multiplexer is defined by the previous recursion. Q_t represents the workload at the end of the time slot t, A_t, represent the amount of traffic (packets) that arrives at the multiplexer, and C_t represents the amount of traffic that is served during the slot t (we do not assume that A_t, C_t are integer valued). Note that the recursion (3) represents the workload of the $G/G/1$ queue with C_t being interpreted as the customer interarrival time,

A_t as the customer service requirements, and Q_t as the queue workload.

Let $X_t = A_t - C_t, t \geq 0$, and let X_t be a sequence of i.i.d. random variables with a distribution F, and A_t independent of C_t. Further, denote its *integrated tail* as $\hat{F}(x) \overset{def}{=} \int_x^\infty [1 - F(t)]dt$. Let's define $F_1(x) = m^{-1}(1 - \hat{F}(x))$, where $m = \hat{F}(0)$. Similarly, in the rest of the paper for any d.f. G, we define its corresponding $\hat{G}(x)$ and $G_1(x)$. Then the following result on the $GI/GI/1$ queue was proven by Veraverbeke [19]. Let B be a d.f. of A_t.

Theorem 1 **(i)** $F_1 \in \mathcal{S} \iff B_1 \in \mathcal{S}$ *and* $\lim_{x\to\infty} \frac{\hat{F}(x)}{\hat{B}(x)} = 1$

(ii) $B_1 \in \mathcal{S} \iff \mathbb{P}[Q_t \leq x] \in \mathcal{S}$ *and each implies*

$$\lim_{x\to\infty} \frac{\mathbb{P}[Q_t > x]}{\hat{B}(x)} = \frac{1}{\alpha - \beta}.$$

Some recent results on long tailed asymptotics of a $GI/GI/1$ are given in [1, 20]. (Also, in [1] further motivation is given for the application of long tailed distributions in communication networks.)

The subexponential asymptotic behavior of the queue length distribution of the Markov-modulated $M/G/1$ queue was derived in [4]. Our extension of this result to the Markov-modulated $G/G/1$ queue is presented in the next section.

3 Markov-modulated $G/G/1$ Queue with Subexponential Arrivals

Consider a Markov-modulated random walk with a negative drift and long-tailed right tail. We prove that the ascending ladder height matrix distribution is asymptotically proportional to a long-tailed distribution. Using this result we show that the queue length distribution of a Markov-modulated $G/G/1$ queue with subexponential arrivals is proportional to the integrated tail of the arrival distribution. Preliminaries on the generalization of the Ladder heights approach to the Markov-modulated random walk recently obtained in [2, 3] are presented in the following subsection (we borrow the notation from the same papers).

3.1 Markov-modulated Random Walk and Ladder Heights

Let $\{J_n\}$ be an irreducible aperiodic Markov chain with a finite state space E (say with N elements) and let $\{X_n\}$ be a sequence of real valued random variables. A Markov process $\{(J_n, X_n)\}$ on $E \times \mathbb{R}$ whose transition distribution depends only on the first coordinate is called a Markov-modulated random walk (MMRW). Let $\{(\tilde{J}_n, \tilde{X}_n)\}$ denote the associated reversed process. This process is determined by a set of transition measures $F_{ij}(A) = \mathbb{P}[J_0 = j, X_1 \in A | J_1 = i]$, and let $F = \{F_{ij}\}$ be the corresponding transition matrix measure.

Further, define $\tilde{S}_0 = 0, \tilde{S}_n = \sum_{i=1}^n \tilde{X}_i$,

$$\begin{aligned}
&\tau_+ = \inf\{n > 0 : \tilde{S}_n > 0\}, \\
&G_+(i,j;A) = \mathbb{P}_i[\tilde{J}_{\tau_+} = j, \tilde{S}_{\tau_+} \in A, \tau_+ < \infty], \\
&\|G_+(i,j)\| = G_+(i,j;(0,\infty)), \\
&G_+(A) = \{G_+(i,j;A)\}_{i,j\in E}, \\
&\|G_+\| = \{\|G_+\|\}_{i,j\in E}.
\end{aligned}$$

The convolution of the matrix measure G_+ is naturally extended to

$$\begin{aligned}
G_+^{*2}(i,j) &= \sum_{k\in E} G_+(i,j) * G_+(k,j), \\
G_+^{*2} &= \{G_+^{*2}(i,j)\}_{i,j\in E};
\end{aligned}$$

higher convolution powers are similarly defined.

Then in [3] the following extension of the Pollaczek-Khinchine identity is provided for $\tilde{M} = \sup_{n\geq 0} \tilde{S}_n$.

Theorem 2 $\mathbb{P}_i[\tilde{M} \in A]$ *is the ith component of the vector*

$$\sum_{n=0}^{\infty} G_+^{*n}(A)(I - \|G_+\|)e,$$

where e is the column vector of ones and I is the identity matrix.

Note: A well known relation between the supremum of the reversed random walk and the stationary queue length distribution is $\mathbb{P}_{\mathrm{i}}[\mathrm{Q} \in \mathrm{A}] = \mathbb{P}_{\mathrm{i}}[\tilde{\mathrm{M}} \in \mathrm{A}]$.

Let $\tau_- = \inf\{n \geq 1 : S_n \leq 0\}$, and define

$$G_-(i,j;A) = \mathbb{P}_{\mathrm{i}}[\mathrm{S}_{\tau_-} \in \mathrm{A}, \mathrm{J}_{\tau_-} = \mathrm{j}, \tau_- < \infty],$$
$$^{\#}G_-(i,j) = \frac{\pi_j}{\pi_i} G_-(j,i);$$

(note that in this definition S is not the reversed process). Then, the following Wiener-Hopf identity holds

$$I - F(A) = (I - {}^{\#}G_-) * (I - G_+)(A), \quad (4)$$

where F is a matrix measure induced by the random walk, and A is any real Borel set. Note that this equation may be difficult to solve. Discussion and references to the computational aspects of the Wiener-Hopf factorization are given in [15].

3.2 Long Tailed Asymptotics of Signed Measures

In this section we prove a few general results on the long tailed asymptotics of signed measures. Combination of these results essentially will give a proof of our main results presented in the following subsection. (These results may also be of independent interest.) Let $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$ be a Borel σ-algebra on $\mathbb{R}$.

Lemma 4 *Let μ, μ_- be two* finite *(signed) measures on $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$, such that $\lim_{x\to\infty} \mu([x,\infty))/\bar{Y}(x) = c$, $Y(x) \in \mathcal{L}$, $|c| < \infty$, and μ_- has support on $(-\infty, 0]$. Then, $\nu \overset{def}{=} \mu_- * \mu$ satisfies*

$$\lim_{x\to\infty} \frac{\nu([x,\infty))}{\bar{Y}(x)} = c\mu_-((-\infty,0]).$$

Proof: Given in [15].

Lemma 5 *Let μ, μ_-, μ_+, be finite (possibly signed) measures on $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$; with μ_- having a support on $(-\infty, 0]$, $|\mu_-((-\infty,0])| > 0$; μ_+ has a support on $[0,\infty)$ and is strictly positive on $[K,\infty)$, $K > 0$, and $\lim_{x\to\infty} \mu([x,\infty))/\bar{Y}(x) = c$, $Y(x) \in \mathcal{L}$, $|c| < \infty$. If $\mu = \mu_- * \mu_+$, then*

$$\lim_{x\to\infty} \frac{\mu_+([x,\infty))}{\bar{Y}(x)} = \frac{c}{\mu_-((-\infty,0])}. \quad (5)$$

Proof: Given in [15]. ◇

Proposition 1 *Let $\mu = \mu_- * \mu_+$, where measures μ, μ_-, μ_+ satisfy the conditions from the previous lemma, and, in addition, $\mu_-((-\infty,0]) = 0$, and $0 < \left|\int_{(-\infty,0]} u d\mu_-(u)\right| < \infty$. Then,*

$$\lim_{x\to\infty} \frac{\mu_+([x,\infty))}{\hat{Y}(x)} = \frac{c}{\int_{(-\infty,0]} u d\mu_-(u)},$$

(recall $\hat{Y}(x) = \int_{[x,\infty)} \bar{Y}(x)$).

Proof: Let $\mu^1([x,\infty)) \overset{def}{=} \int_x^\infty \mu([u,\infty))du$, and $\mu_-^1([z,0]) \overset{def}{=} \int_{[z,0]} \mu_-([u,0])du$. Observe that $\mu^1([x,\infty)) \sim c\hat{Y}(x)$ as $x \to \infty$. Also, from the assumptions it follow that μ_-^1 defines a finite measure on $(-\infty,0]$, since $\mu_-^1((-\infty,0]) = \int_{-\infty}^0 -\mu_-((-\infty,u))du = \int_{(-\infty,0]} u\mu_-(du)$. Then, by applying Fubini's theorem (see [21], pp. 180), we get

$$\begin{aligned}
&\mu^1([y,\infty)) \\
&= \int_y^\infty du \int_{[0,\infty)} \mu_-([u-x,0])\mu_+(dx) \\
&= \int_{[0,\infty)} \mu_+(dx) \int_y^x \mu_-([u-x,0])du \\
&= \int_{[0,\infty)} \mu_-^1([y-x,0])\mu_+(dx).
\end{aligned}$$

So, $\mu_1([y,\infty))$ is obtained by convolution of finite measures μ_+, μ_-^1. Therefore, by applying lemma 5, the conclusion of the proposition follows (recall that $\mu_-^1((-\infty,0]) = \int_{(-\infty,0]} u\mu_-(du)$).

◇

3.3 Subexponential Queue Asymptotics

We will now proceed to prove our main result of this section. In order to state the result we need to introduce some additional notation. Let $H(x) = \{H_{ij}\}$

be a matrix composed of distribution functions, and its Fourier transform defined as $\tilde{H}(\omega) = \{\tilde{H}_{ij}(\omega)\}$, $\tilde{H}_{ij}(\omega) = \int_{-\infty}^{\infty} e^{i\omega x} dH_{ij}(x)$. We will use the symbol $\mathcal{F}^{-1}(\cdot)$ to denote the operation of taking the inverse Fourier transform. Note that there is a one-to-one correspondence between the distribution functions on $\mathbb{R}$ and its Fourier transforms (see [21], section 8.3). The Wiener-Hopf factorization can be written as

$$(I - \tilde{F}(\omega)) = (I - {}^{\#}\tilde{G}_{-}(\omega))(I - \tilde{G}_{+}(\omega)). \tag{6}$$

Observe also that $\tilde{H}_{ij}(0) = H((-\infty, \infty))$, and that $-i\tilde{H}'_{ij}(0) = \int_{-\infty}^{\infty} x dH_{ij}(x)$; assume that the first moments of H_{ij} are finite. For any matrix A let us define the *adjoint* matrix $adj(A)$, $adj(A)_{ij} = (-1)^{i+j}\det(A^{ij})$, where A^{ij} denotes the matrix obtained by deleting the ith row and jth column from A. If A is invertible then $A^{-1} = (\det(A))^{-1} adj(A)$. Assume $\mathbb{E}\mathrm{X_n} < 0$ (negative drift) and $|\int_{-\infty}^{\infty} x F_{ij}(dx)| < \infty$.

Theorem 3 *Let $\bar{F}(x)/\bar{Y}(x) \sim W$, $W = \{W_{ij}\}$, $W_{ij} < \infty$, $Y(x) \in \mathcal{L}$. Then,*

$$\frac{\bar{G}_{+}(x)}{\bar{Y}(x)} \sim \frac{W adj(I - {}^{\#}\tilde{G}_{-})(0)}{-i\det(I - {}^{\#}\tilde{G}_{-})'(0)}, \tag{7}$$

as $x \to \infty$.

Proof: Fist let us observe that $\det(I - \tilde{F})(\omega)$ has a zero of order one for $\omega = 0$. $\omega = 0$ is a zero since $\det(I - \tilde{F})(0) = \det(I - P) = 0$, as P is a stochastic transition matrix; that this zero is of order one follows from $\mathbb{E}\mathrm{X_n} < 0$. Furthermore, since $\|G_{+}\|$ is substochastic (see [2], proposition 4.2), we have that $|\det(I - \tilde{G}_{+})(0)| > 0$, which implies (by equation (6)) that $\omega = 0$ is also a zero of order one for the $\det(I - {}^{\#}\tilde{G}_{-})(\omega)$, implying $0 < |\det(I - {}^{\#}\tilde{G}_{-})'(0)| < \infty$; finiteness follows from $\det(I - \tilde{F})'(0) = \det(I - \tilde{G}_{+})(0)\det(I - {}^{\#}\tilde{G}_{-})'(0)$.

Let us define the measure $\mu_{-} \overset{def}{=} \mathcal{F}^{-1}(\det(I - {}^{\#}\tilde{G}_{-}(\omega)))$. Note that this measure has support on $(-\infty, 0]$ and $\int_{(-\infty,0]} u d\mu_{-}(u) = -i\det(I - {}^{\#}\tilde{G}_{-})'(0) \neq 0$ (finite). Also, equation 6 can be written as

$$\begin{aligned}(\tilde{F}(\omega) - I) & adj(I - {}^{\#}\tilde{G}_{-}(\omega)) \\ &= (\tilde{G}_{+}(\omega) - I)\det(I - {}^{\#}\tilde{G}_{-}(\omega)),\end{aligned}$$

or componentwise

$$\begin{aligned}(\tilde{G}_{+}(\omega) - I)_{ij} & \det(I - {}^{\#}\tilde{G}_{-}(\omega)) \\ &= \sum_{k=1}^{n} (\tilde{F}(\omega) - I)_{ik} adj(I - {}^{\#}\tilde{G}_{-}(\omega))_{kj}.\end{aligned}$$

If $\mu_{ij} \overset{def}{=} \mathcal{F}^{-1}(\sum_{k=1}^{n}(\tilde{F}(\omega) - I)_{ik} adj(I - {}^{\#}\tilde{G}_{-}(\omega))_{kj})$, then, by Lemma 4, $\mu([x, \infty)) \sim \sum_{k=1}^{n} W_{ik} adj(I - {}^{\#}\tilde{G}_{-}(0))_{kj}\bar{Y}(x)$ as $x \to \infty$. If $\mu_{+ij} \overset{def}{=} \mathcal{F}^{-1}(\tilde{G}_{+}(\omega) - I)_{ij}$, then $\mu_{ij} = \mu_{+ij} * \mu_{-}$, where $\mu_{ij}, \mu_{+ij}, \mu_{-}$, satisfy the conditions of proposition 1. Thus, by the same proposition,

$$\frac{\bar{G}_{+ij}(x)}{\hat{Y}(x)} \sim \frac{\sum_{k=1}^{n} W_{ik} adj(I - {}^{\#}\tilde{G}_{-}(0))_{kj}}{-i\det(I - {}^{\#}\tilde{G}_{-})'(0)},$$

which is tantamount to equation (7). This finishes the proof of the theorem. ◇

Now by combining the previous theorem and the extension of lemma 3 from [4] we will prove the following theorem on subexponential asymptotics for the Markov-modulated $G/G/1$ queue. Let (J_n, A_n) and (J_n, C_n) be two MMRW such that (J_n, A_n) and (J_n, C_n) are conditionally independent given J_{n-1}, J_n. $\{A_n\}$ and $\{C_n\}$ are arrival and service processes, respectively. Let B be a reverse transition matrix measure of the MMRW (J_n, A_n), such that for each $i, j \in E$ $\bar{B}_{ij}(x) \sim W_{ij}\bar{Y}(x)$ as $x \to \infty$ and $W_{ij} > 0$.

Theorem 4 *If $Y_1 \in \mathcal{S}$ then there exists a constant $c > 0$ such that.*

$$\frac{\mathbb{P}[\mathrm{Q_t > x}]}{\int_{\mathrm{x}}^{\infty} \mathbb{P}[\mathrm{A_t > u}]\mathrm{du}} \sim c \text{ as } x \to \infty.$$

Furthermore, when the Wiener-Hopf factorization is explicitly solvable the constant c is explicitly computable.

Proof: Componentwise the asymptotic proportionality of the matrix distributions $\hat{F}(x)$ and $\hat{B}(x)$ follows from Theorem 1 (i) as $x \to \infty$. Then, combining Theorems 2, 3, and Lemma 3 the conclusion of the theorem follows. ◇

Remark: The assumption $W_{ij} > 0$ for all i, j can be removed. The queue distribution is subexponential as long as at least one measure B_{ij} is subexponential. For a precise statement of this result see [15].

Illustration of this remark and the preceding theorem is given in the following numerical example.

Example 2 Consider a constant server queue with $C_t = 1$ and two state (say $\{0,1\}$) Markov-modulated arrivals (source). The transition probabilities for the modulating Markov chain are $p_{01} = 1/3, p_{10} = 3/4$. When in state 0 the source is producing zero arrivals, and when in state 1 the source is producing (independent of the previous state) arrivals according to the distribution $\mathbb{P}[\mathrm{A_t} = 0|\mathrm{J_t} = 1] = 0.327144, \mathbb{P}[\mathrm{A_t} = 1|\mathrm{J_t} = 1] = 0$, and $\mathbb{P}[\mathrm{A_t} = \mathrm{i}|\mathrm{J_t} = 1] = \mathrm{w}/\mathrm{i}^5, \mathrm{w} = 18.220859, 2 \leq \mathrm{i} \leq 350$. (Note that these are bounded arrivals.) Thus, according to the previous theorem (and the remark after it) the queue length distribution is proportional to $1/i^4$. The comparison between the true probabilities and the approximation $c/i^4, c = 2.617872$ is given in figure 2.

4 Asymptotics of a Fluid Flow Queue with Subexponentially Correlated Arrivals

In this section we construct a class of processes for which we show that its autocorrelation function (ac.f.) is subexponential. Furthermore, when these processes are fed to a fluid flow queue, we prove the asymptotic proportionality of the queue length distribution with the arrival process ac.f.. Throughout this section we assume a continuous time model (of course all the results are valid for discrete time also).

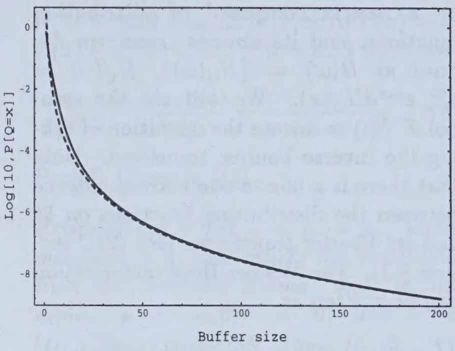


Figure 2: Graph of $\log_{10} \mathbb{P}[\mathrm{Q} = \mathrm{i}]$ versus buffer size i from Example 2; solid line represents the true probabilities, and dashed line represents the approximation $2.617872/i^{-4}$.

4.1 Stationary Subexponentially Correlated Arrivals

Consider a point process $T = \{T_0 \leq 0, T_n, n \geq 1\}$ such that $T_n - T_{n-1}, n \geq 1$ are i.i.d. with subexponential distribution function F. Further, let $J_n, n \geq 0$ be an irreducible aperiodic Markov Chain with finite state space $\{1, \ldots, K\}$, transition matrix $\{P_{ij}\}$, and stationary probability distribution $\pi_i, 1 \leq i \leq K$. In order to make this process stationary (see [9], section 9.3), we choose the residual time at zero until the first jump to be distributed as an integrated tail of F, i.e., $F_1(t) = \mathbb{P}[\mathrm{T_1} \leq \mathrm{t}] = \mathrm{m}^{-1} \int_{0,\mathrm{t}} \bar{\mathrm{F}}(\mathrm{u})\mathrm{du}$.

Now we construct the following process:

$$A_t = J_n \quad \text{for} \quad T_{n-1} \leq t < T_n. \qquad (8)$$

called a Markov Chain Embedded in a Stationary Subexponential Renewal Process (MCESSR). A typical sample path of this process is given in figure 3. It is well known that under fairly general

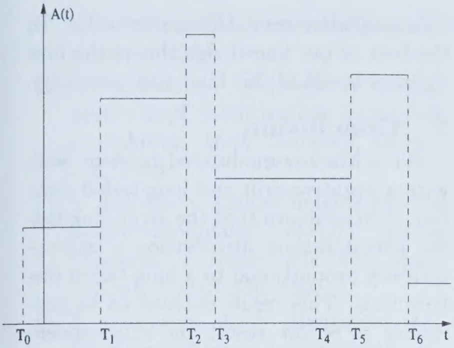


Figure 3: A possible realization of a Markov chain embedded into a renewal process.

conditions a Markov chain converges exponentially fast to the steady state distribution. However, the process that we have constructed, because of the subexponentially distributed sojourn times, converges with subexponential speed to its steady state. This is stated in the following lemma.

Proposition 2 *If $F_1 \in \mathcal{S}$, then*

$$(\mathbb{P}_i[A_t = j] - \pi_j)\bar{F}_1(t)^{-1} \sim (\delta_{ij} - \pi_j),$$

as $t \to \infty$, and $\delta_{ij} = 1$ if $i = j$ and zero otherwise.

Proof: Given in [15] ◇

We will illustrate this lemma by the following example.

Example 3 Let F be a discrete distribution function with support $[1, 1000]$, $\mathbb{P}[T_2 - T_1 = 1] = 0.186532$, and $\mathbb{P}[T_2 - T_1 = i] = w/i^5, w = 22.028625$, $2 \leq i \leq 1000$; chose a two state Markov chain with transition probabilities $p_{01} = 1/3$ and $p_{10} = 3/4$. Then, the functions $(d_{i,1}(t) \overset{def}{=} (\mathbb{P}_i[A_t = 1] - \pi_1)(\bar{F}_1(t)(\delta_{i1} - \pi_1))^{-1}$, $i = 0, 1$, converge to one as $t \to \infty$, with subexponential rate. This can be clearly seen in figure 4.

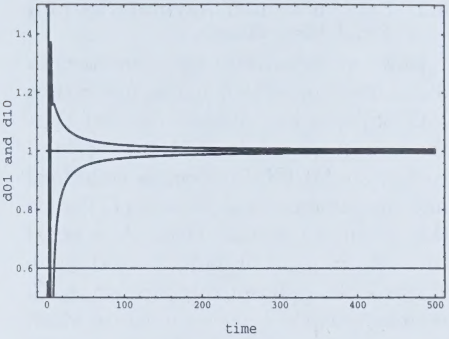


Figure 4: Functions $d_{i,1}(t) \overset{def}{=} (\mathbb{P}_i[A_t = 1] - \pi_1)(\bar{F}_1(t)(\delta_{i1} - \pi_1))^{-1}$, $i = 0, 1$. The graph shows that $d_{i,1}(t) \to 1$ as $t \to 1$.

Now it is easy to prove that the covariance function $\rho(t) \overset{def}{=} (\mathbb{E}A_0A_t - (\mathbb{E}A_0))^2$ of the MCESSR process satisfies the following asymptotic relation. Let $\mathrm{Var}(J_0) > 0$ be the variance of J_0.

Theorem 5 *If $F_1 \in \mathcal{S}$, then*

$$\rho(t) \sim \mathrm{Var}(J_0)\bar{F}_1(t),$$

as $t \to \infty$.

Remark: A nontrivial extension of this theorem and the previous lemma for the case of subexponential semi-Markov processes is given in [15].

Proof: By applying the previous lemma and after some simple algebraic manipulations we get

$$\begin{aligned} &\left(\mathbb{E}A_0A_t - (\mathbb{E}A_0)^2\right) \bar{F}_1(t)^{-1} \\ &= \sum_{i,j} a_i a_j (\pi_i \mathbb{P}_i[A_t = j] - \pi_i\pi_j)\bar{F}_1(t)^{-1} \\ &\sim \sum_{i,j} a_i a_j \pi_i (\delta_{ij} - \pi_j) \\ &= \sum_i \pi_i a_i^2 - \sum_{i,j} \pi_i\pi_j a_i a_j = \mathrm{Var}(J_0), \end{aligned}$$

as $t \to \infty$. This completes the proof of the theorem. ◇

4.2 Subexponential Asymptotics of a Fluid Flow Queue

Now, we investigate the queue length distribution of a fluid queue fed with a MCESSR process. We assume that both the arrival process A_t and the server process C_t are MCESSR processes embedded into the same renewal process $\{T_n\}$, such that when the Markov chain J_n is equal to i, $A_t = a_i \geq 0$ and $C_t = c_i \geq 0$, $T_n \leq t < T_{n+1}$. Intuitively the pair A_t, C_t represents a fluid queueing model in which $A_t = a_i$ means that the flow is arriving to the queue with a rate a_i and $C_t = c_i$ means that the flow is coming from the queue with a rate c_i. We will calculate the queue length distribution at the jump times (Palm probability); $Q_n \equiv Q(T_n)$ satisfies the recursion

$$Q_{n+1} = (Q_n + x_{J_n} \Delta T_n)^+,$$

where $x_i = a_i - c_i$, and $\Delta T_n = T_{n+1} - T_n$.

We are now ready to state the following result on the asymptotic proportionality of the queue length distribution and its autocorrelation function. To avoid trivialities we assume that at least for one i, $x_i > 0$; also $Var(J_0) > 0$.

Theorem 6 *Let the stability condition* $\mathbb{E}x_{J_n} < 0$ *be satisfied, and for all* $x_i > 0$, $\mathbb{P}[\Delta T_n > t/x_i]/\bar{Y}(t) \sim w_i$, *as* $x \to \infty$, *with at least one* $w_i > 0,$, *and* $Y_1 \in \mathcal{S}$. *Then, there exist a positive constant* r *such that*

$$\mathbb{P}[Q > t] \sim r\ \rho(t).$$

Proof: Follows by straightforward combination of theorems 4, 5; (more precisely an extension of theorem 4 mentioned in the remark after it). ◇

Remarks: (i) If the distribution function of ΔT_n belongs to the Pareto family the assumption $\mathbb{P}[\Delta T_n > t/x_i] \sim w_i \bar{Y}(t), w_i > 0$ will be satisfied for all $x_i > 0$. **(ii)** Taking into account the remark after the theorem 5 we see that this relationship holds for the general class of *subexponential semi-Markov* arrivals. To the best of our knowledge this is the first rigorous result of this kind and generality.

5 Conclusion

For a Markov-modulated random walk with a negative drift and long-tailed right tail we have shown that the ascending ladder height matrix distribution is asymptotically proportional to a long-tailed distribution. This result enabled us to generalize a recent result on subexponential asymptotics of a Markov-modulated M/G/1 queue to subexponential asymptotics of a Markov-modulated G/G/1 queue. If the matrix Wiener-Hopf factorization is explicitly solvable then the asymptotic constant of proportionality is explicitly computable.

We also constructed a general class of processes, termed MCESSR, for which the covariance (autocorrelation) function has a subexponential tail. Furthermore, when this processes are fed into a fluid flow queue, the queue length distribution was proven to be asymptotically *proportional to its autocorrelation function.* In short

- (subexp. marginal d.f. + exp. ac.f.) ⇒ (queue distribution is determined by marginal d.f.),
- (bounded (exp) marginal d.f. + subexp. ac.f.) ⇒ (queue distribution is determined by ac. f.).

When these type of conditions are met in practice, the above results may lead to an efficient admission control policy at network multiplexers. Admission controllers may decide its admission control policy based on either the marginal distributions or the autocorrelation functions of the arrival streams, depending which conditions are satisfied.

References

[1] J. Abate, G.L. Choudhury, and W. Whitt. Waiting-time tail probabilities in queues with long-tail

service-time distributions. *Queueing systems.*, 1994.

[2] S. Asmussen. Aspects of matrix wiener-hopf factorization in applied probability. *Math. Scientist*, 14:101–116, 1989.

[3] S. Asmussen. Ladder heights and the markov-modulated M/G/1 queue. *Stochastic Processes and their Applications*, 37:313–326, 1991.

[4] S. Asmussen, L. F. Henriksen, and C. Klüppelberg. Large claims approximations for risk processes in a markovian environment. *Stochastic Processes and their Applications*, 54:29–43, 1994.

[5] K. B. Athreya and P. E. Ney. *Branching Processes.* Springer-Verlag, 1972.

[6] C. S. Chang. Stability, queue lenght and delay of deterministic and stochastic queueing networks. *IEEE Transactions on Automatic Control*, 39:913–931, 1994.

[7] V. P. Chistakov. A theorem on sums on independent positive random variables and its application to branching random processes. *Theor. Probab. Appl.*, 9:640–648, 1964.

[8] Gagan L. Choudury, David M. Lucantoni, and Ward Whitt. Squeezing the most of atm. *to appear in IEEE Trans. on Communications*, 1995.

[9] E. Cinlar. *Introduction to Stochastic Processes.* Prentice-Hall, 1975.

[10] A. I. Elwalid and D. Mitra. Effective bandwidth of general markovian traffic sources and admission control of high speed networks. *IEEE/ACM Trans. on Networking*, 1(3):329–343, June 1993.

[11] M. W. Garett and W. Willinger. Analysis, modeling and generation of self-similar vbr video traffic. In *SIGCOMM'94*, pages 269–280, 1994.

[12] P. V. Glynn and W. Whitt. Logarithmic asymptotics for steady-state tail probabilities in a single-server queue. *Studies in Appl. Prob.*, 1994.

[13] R. Guerin, H. Ahmadi, and M. Nagshineh. Equivalent capacity and its application to bandwidth allocation in high-speed networks. *IEEE J. Select. Areas Commun.*, 9:968–981, 1991.

[14] P. R. Jelenković and A. A. Lazar. On the dependence of the queue tail distribution on multiple time scales of atm multiplexers. In *Conference on Information Sciences and Systems*, pages 435–440, Baltimore, MD, March 1995.

[15] P. R. Jelenković and A. A. Lazar. Subexponential asumptotics of a network multiplexer. CTR Technical Report CU/CTR/TR, Columbia University, September 1995.

[16] P. R. Jelenković and A. A. Lazar. Evaluating the queue length distribution of an atm multiplexer with multiple time scale arrivals. In *Proceedings of INFOCOM'96*, San Francisco, California, March 1996, *to appear.*

[17] F. P. Kelly. Effective bandwidths at multi-class queues. *Queueing Systems*, 9:5–16, 1991.

[18] D. Tse, R. Gallager, and J. Tsitsiklis. Statistical multiplexing of multiple time-scale markov streams. *IEEE, Selected Areas in Communications*, August 1995.

[19] N. Veraverbeke. Asymptotic behavior of wiener-hopf factors of a random walk. *Stochastic Proc. and Appl.*, 5:27–37, 1977.

[20] E. Willekens and J. L. Teugels. Asymptotic expansion for waiting time probabilities in an M/G/1 queue with long-tailed service time. *Queueing Systems*, 10:295–312, 1992.

[21] Chow Y. S. and Teicher H. *Probability Theory.* Springer-Verlag, 1988.

ASYMPTOTIC BEHAVIOR OF A RELAXED SEQUENTIAL FLOW-CONTROL ALGORITHM FOR MULTICLASS NETWORKS

Christopher M. Rump
Department of Industrial Engineering
State University of New York at Buffalo
Buffalo, NY 14260-2050, USA

Shaler Stidham, Jr.
Department of Operations Research
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599-3180, USA

Abstract

A net-benefit (throughput reward less delay cost) criterion for optimal flow control in a multiclass network is investigated. A relaxed sequential algorithm for obtaining a Nash equilibrium is proposed in which network users iteratively update (in turn) their throughput strategy in response to local information on the strategies of other users. The novelty of this flow-control algorithm is in its relaxation, whereby each user employs a strategy update that is a convex combination of his previous strategy and his best-reply strategy to current information. Alternatively, this relaxation can be viewed as an exponential smoothing of all previous best-reply strategies. This algorithm is shown to converge asymptotically for any number of user classes over an interval of the relaxation parameter. Additionally, in the case of two classes, relaxation is shown to accelerate convergence to equilibrium and can create a Hopf bifurcation at which the algorithm neither converges nor diverges. Due to an equivalent functional form, these results can be directly translated to a network of users employing a power criterion.

1 Introduction

Consider a wide-area network that is shared by m customer classes. These customer classes are distinguished by differences in their performance objective, reflected by the cost associated with service delay and the value placed on receiving that service. In this multiclass environment, an efficient allocation of resources can be achieved by allowing the flow rate from each class i, $i = 1, \ldots, m$, along a particular link in the network to be regulated by a class manager. When this flow control is performed using only local information concerning the utilization of that link, there is no need for a central controller, and hence we may consider each link separately from the rest of the network.

Several studies have investigated rate-based algorithms for implementing this decentralized flow control (for a survey see [5]). At each step of these algorithms, a user updates their throughput strategy based on new information received about the strategies of other users of the link. Typically, this update optimizes some performance objective and constitutes the best-reply to the current information.

However, since the strategies of other users are being updated as well, best-reply is a somewhat myopic policy. It seems worthwhile to investigate the effect of using historical as well as current information in the strategy update. We propose an exponential smoothing (cf., e.g., [1]), i.e., geometric weighting, of the entire history of best-reply strategies as a simple and plausible means to do so. An exponential smoothing reduces the update to a convex combination of the current strategy and the new best-reply, which we call the algorithmic *relaxation*.

The term relaxation is borrowed from the field of distributed computing in which the study of relaxed matrix-iterative algorithms for solving a linear system are motivated by the desire to accelerate the rate of convergence to a solution. We are similarly motivated by the desire to achieve equilibrium convergence as quickly as possible. Although not directly incorporated as a cost to the user, it is certainly undesirable to bear the stress of a system that is out of equilibrium. The quicker that the system converges to equilibrium, the better.

In the next section, we present the queueing model and net-benefit criteria used by network users. Section 3 introduces the relaxed sequential algorithm to be studied. (An asynchronous implementation will be studied in the sequel [7].) A unique Nash equilibrium solution to the system of nonlinear net-benefit objectives is found in Section 4. In Section 5 we linearize the problem to allow investigation of the asymptotic behavior of the relaxed sequential algorithm in a neighborhood of the equilibrium, and then show in Section 6 that this linearized form is equivalent to a power-criterion model. In Section 7 we analyze convergence of the algorithm and then focus in Section 8 on acceleration to and bifurcation of the equilibrium for the special case of two user classes. Section 9 concludes the paper.

2 The Model

We model the link as an *M/GI/1* queue. Arriving customer jobs (or packets) from class i form a Poisson process with rate λ_i. These jobs are then serviced by a single server with service rate μ. The sizes of customer service requests are i.i.d., distributed as a random variable Y with first and second moments $E[Y] = 1$ and $E[Y^2] = \beta \geq 1$, respectively, independent of class. Treating the service rate μ as a scale parameter, the service times are therefore distributed as Y/μ so that the mean and variance of service times are $1/\mu$ and $(\beta - 1)/\mu^2$, respectively. The FCFS (no priority) service discipline and a common distribution of service times for all customer classes then ensures that the waiting time in the system is class independent.

Within each user class i there is a maximum potential admission rate Λ_i and, of course, a minimum rate of 0. We require that the aggregate throughput of all users, denoted by $\lambda = \sum_{j=1}^{m} \lambda_j$, satisfy the queue-stability constraint $\lambda < \mu$. Thus, for each user class i we define an admissible space $\mathcal{L}_i = \{\lambda_i | \; 0 \leq \lambda_i \leq \min\{\Lambda_i, \bar{\lambda}_i - \epsilon\}\}$, where $\bar{\lambda}_i = \mu - \sum_{j \neq i} \lambda_j$ is the capacity available to user class i, and $\epsilon > 0$ is some arbitrarily small constant. The simultaneous admissible space of all user classes $1, \ldots, m$ is then the coupled constraint set $\mathcal{L} = \{\, \boldsymbol{\lambda} | \; \mathbf{0} \leq \boldsymbol{\lambda} \leq \boldsymbol{\Lambda}, 0 \leq \lambda \leq \mu - \epsilon\}$, where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_m)$ is the vector of throughputs, and $\boldsymbol{\Lambda} = (\Lambda_1, \ldots, \Lambda_m)$ is the vector of maximal throughputs.

Class-i customers are assumed to have a linear delay cost $h_i > 0$ per unit time spent waiting in the system. Upon completion of service, each customer then receives a fixed reward r_i. The objective of manager i is to maximize the long-run average net benefit (throughput reward less delay cost) per unit time for jobs in class i, namely,

$$\max_{\lambda_i \in \mathcal{L}_i} B_i(\boldsymbol{\lambda}) = \max_{\lambda_i \in \mathcal{L}_i} \{r_i \lambda_i - h_i L_i(\boldsymbol{\lambda})\}, \qquad i = 1, \ldots, m,$$

where $L_i(\boldsymbol{\lambda}) = \lambda_i W(\lambda)$ is the expected number of class-i users in the system, and $W(\lambda)$ is the expected waiting time in the system. For an *M/GI/1* queueing system, the expected delay can be derived from the Pollaczek-Khintchine (P-K) mean-value formula (cf., e.g., [2]), yielding for $\beta' = 1 - \beta/2 \leq 1/2$

$$W(\lambda) = (\mu - \beta'\lambda)\,[\mu\,(\mu - \lambda)]^{-1}. \tag{1}$$

Since $B_i(0,\ldots,0,\lambda_i,0,\ldots,0) = \lambda_i(r_i - h_i W(0,\ldots,0,\lambda_i,0,\ldots,0)) \leq \lambda_i(r_i - h_i/\mu)$, we assume that for each class i the reward r_i exceeds the expected delay cost h_i/μ upon entering an empty system.

The optimal strategy for player i, given fixed strategies of the other players, is

$$\gamma_i = \arg\max_{\lambda_i \in \mathcal{L}_i} B_i(\lambda_1,\ldots,\lambda_i,\ldots,\lambda_m) = \Gamma_i(\lambda_1,\ldots,\lambda_i,\ldots,\lambda_m).$$

for some function $\Gamma_i(\cdot)$ of the other users' strategies. The Nash equilibria $\boldsymbol{\lambda}^* \in \mathcal{L}$ at which no player has an incentive to unilaterally deviate from his current strategy must therefore satisfy

$$\lambda_i^* = \gamma_i^* = \arg\max_{\lambda_i \in \mathcal{L}_i} B_i(\lambda_1^*,\ldots,\lambda_i,\ldots,\lambda_m^*), \qquad i = 1,\ldots,m, \tag{2}$$

and constitute fixed point solutions to $\boldsymbol{\lambda} = \boldsymbol{\Gamma}(\boldsymbol{\lambda})$.

3 A Relaxed Sequential Algorithm

To achieve a Nash equilibrium among the network users, we propose an iterative flow-control algorithm, operating over a discrete time sequence $n = 0, 1, \ldots$, in which users take turns (in sequence) in updating their flow-control strategies. The sequence is not unique and any ordering will suffice provided that it is kept throughout. After a particular user adjusts his strategy to optimize the throughput versus delay trade-off, the value of the new rate is broadcast to all the other users. The next user in the sequence then proceeds to update his strategy. This step-by-step algorithm is called the *sequential algorithm.*

At step $n + 1 \geq 1$ of the sequential algorithm the optimal (best-reply) strategy for user $i = 1,\ldots,m$ is given by

$$\gamma_i^{(n)} \;=\; \arg\max_{\lambda_i \in \mathcal{L}_i} B_i(\lambda_1^{(n+1)},\ldots,\lambda_{i-1}^{(n+1)},\lambda_i,\lambda_{i+1}^{(n)},\ldots,\lambda_m^{(n)})$$

with arbitrary but admissible initial strategy vector $\boldsymbol{\lambda}^{(0)} = (\lambda_1^{(0)},\ldots,\lambda_m^{(0)})$. However, we shall suppose that each user uses a *relaxed* update which is a convex combination of his previous strategy $\lambda_i^{(n)}$ and the new optimal strategy $\gamma_i^{(n)}$, viz,

$$\lambda_i^{(n+1)} = (1-\omega)\lambda_i^{(n)} + \omega\gamma_i^{(n)}, \tag{3}$$

where ω is an arbitrary relaxation parameter. Notice that when $\omega = 1$ there is no relaxation and the procedure reduces to best-reply. The case $\omega < 1$ ($\omega > 1$) is called underrelaxation (overrelaxation). The recursive relation (3) is easily recognized as an exponential smoothing of the past history of best-reply strategies $\gamma_i^{(0)}, \gamma_i^{(1)}, \ldots, \gamma_i^{(n)}$ where $\gamma_i^{(0)} \equiv \lambda_i^{(0)}$.

4 The Interior Nash Equilibrium

The necessary and sufficient Karush-Kuhn-Tucker (KKT) optimality conditions for an interior Nash equilibrium $\boldsymbol{\lambda}^* > \mathbf{0}$ satisfying (2) (achieved with optimal multipliers $\boldsymbol{u}_i^* = \mathbf{0}$ for all i) are

$$\nabla_i B_i(\boldsymbol{\lambda}) = r_i - h_i \nabla_i L_i(\boldsymbol{\lambda}) = 0, \qquad i = 1, \ldots, m. \tag{4}$$

Notice that for each $i = 1, \ldots, m$

$$\begin{aligned}
\nabla_i L_i(\boldsymbol{\lambda}) &= W(\lambda) + \lambda_i \nabla_i W(\lambda) = \frac{\mu - \beta'\lambda}{\mu(\mu - \lambda)} + \frac{(1-\beta')\lambda_i}{(\mu - \lambda)^2} \\
&= \frac{(\mu - \beta'\lambda)(\mu - \lambda) + (1-\beta')\mu\lambda_i}{\mu(\mu - \lambda)^2} \\
&= \frac{((1-\beta')\mu + \beta'\bar{\lambda}_i - \beta'\lambda_i)(\bar{\lambda}_i - \lambda_i) + (1-\beta')\mu\lambda_i}{\mu(\bar{\lambda}_i - \lambda_i)^2} \\
&= \frac{\beta'\lambda_i^2 - 2\beta'\bar{\lambda}_i\,\lambda_i + ((1-\beta')\mu + \beta'\bar{\lambda}_i)\bar{\lambda}_i}{\mu(\bar{\lambda}_i - \lambda_i)^2}.
\end{aligned}$$

Substituting this expression into (4) yields for $i = 1, \ldots, m$ the equilibrium condition

$$h_i(\beta'\lambda_i^2 - 2\beta'\bar{\lambda}_i\,\lambda_i + ((1-\beta')\mu + \beta'\bar{\lambda}_i)\bar{\lambda}_i) = r_i\mu(\bar{\lambda}_i - \lambda_i)^2. \tag{5}$$

By combining terms, (5) reduces to $h_i(1-\beta')\mu\;\bar{\lambda}_i = (r_i\mu - h_i\beta')(\bar{\lambda}_i - \lambda_i)^2$. Since $r_i\mu - h_i\beta' > 0$ follows from the assumption $r_i\mu > h_i$ and the fact that $\beta' \le 1/2$, we divide through by this positive quantity, leaving the following nonlinear system of m equations in m unknowns:

$$\nu_i\bar{\lambda}_i = (\bar{\lambda}_i - \lambda_i)^2, \qquad i = 1, \ldots, m, \tag{6}$$

where $\nu_i = (h_i(1-\beta')\mu)/(r_i\mu - h_i\beta') > 0$. To solve such a system, first notice that for any solution vector the right hand sides of (6) are all the same, namely, $(\bar{\lambda}_i - \lambda_i)^2 = (\mu - \lambda)^2$. Thus, for some unknown ν we can rewrite (6) as

$$\nu_i\bar{\lambda}_i = \nu^2, \qquad i = 1, \ldots, m. \tag{7}$$

To impose the definition of ν, we add an additional equation

$$\mu - \lambda = \nu, \tag{8}$$

and view $\nu \in (0, \mu]$ as the "excess" capacity. Temporarily treating μ as an unknown and ν as known, we combine (7) and (8) to produce a linear system of $m+1$ equations in $m+1$ unknowns given by $\boldsymbol{A}(\mu, \boldsymbol{\lambda})^{\mathrm{T}} = \boldsymbol{b}$, with

$$\boldsymbol{A} = \begin{pmatrix} 1 & 0 & -1 & \cdots & -1 \\ 1 & -1 & 0 & \cdots & -1 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 1 & -1 & -1 & \cdots & 0 \\ 1 & -1 & -1 & \cdots & -1 \end{pmatrix} \quad \text{and} \quad \boldsymbol{b}^{\mathrm{T}} = \left(\frac{\nu^2}{\nu_1}, \ldots, \frac{\nu^2}{\nu_m}, \nu\right).$$

Solving this system yields the unique interior equilibrium solution

$$\lambda_i^* = \nu(\nu/\nu_i - 1), \qquad i = 1, \ldots, m \tag{9}$$

and the relation

$$\mu = \nu^2 \sum_{j=1}^{m} \nu_j^{-1} - (m-1)\nu. \tag{10}$$

Since μ is known we now solve the quadratic (10) in $\nu > 0$ yielding

$$\nu = \left(2\sum_{j=1}^{m} \nu_j^{-1}\right)^{-1} \left(m - 1 + \left[(m-1)^2 + 4\mu \sum_{j=1}^{m} \nu_j^{-1}\right]^{1/2}\right). \tag{11}$$

To ensure an interior equilibrium $\lambda_i^* > 0$ in (9), we require $\nu_i < \nu$ for all $i = 1, \ldots, m$.

5 Linearization

In order to study the asymptotic behavior of the sequential algorithm, we begin by examining user i's equilibrium-seeking condition in (6). Since we desire $\lambda_i < \bar{\lambda}_i$ for a stable queueing system, the equilibrium solution for user i is given by $\lambda_i = \bar{\lambda}_i - (\nu_i \bar{\lambda}_i)^{1/2}$. Thus, armed with some recent knowledge of the other users' strategies, the best reply of user i is given by the nonlinear mapping $\Gamma_i(\boldsymbol{\lambda}) = \bar{\lambda}_i - (\nu_i \bar{\lambda}_i)^{1/2}$.

To analyze such a nonlinear system, we appeal to the Liapunov's first method (cf., e.g., [6]), whereby we examine a linearized approximation of the system near the equilibrium fixed point $\boldsymbol{\lambda}^*$ of the difference equation $\boldsymbol{\lambda}^{(n+1)} = \boldsymbol{\Gamma}(\boldsymbol{\lambda}^{(n)})$. The $m \times m$ Jacobian matrix of $\boldsymbol{\Gamma}(\boldsymbol{\lambda})$ has elements $\partial\Gamma_i/\partial\lambda_j = \begin{cases} -1 + (1/2)(\nu_i/\bar{\lambda}_i)^{1/2} & , i \neq j \\ 0 & , i = j \end{cases}$.

Evaluation of the information $\bar{\lambda}_i$ at the equilibrium using (7) yields the Jacobi matrix

$$\boldsymbol{J} = \begin{pmatrix} 0 & -\delta_1 & \cdots & -\delta_1 \\ -\delta_2 & 0 & \cdots & -\delta_2 \\ \vdots & \vdots & \ddots & \vdots \\ -\delta_m & -\delta_m & \cdots & 0 \end{pmatrix},$$

where $\delta_i = 1 - (1/2)(\nu_i/\nu) \in (\frac{1}{2}, 1)$.

Defining a vector of displacements $\boldsymbol{y}^{(n)} = \boldsymbol{\lambda}^{(n)} - \boldsymbol{\lambda}^*$ from the equilibrium point provides a linearized fixed-point equation $\boldsymbol{y} = \boldsymbol{J}\boldsymbol{y}$. This matrix equation can be written as the linear system of equations $\boldsymbol{B}\boldsymbol{y} = \boldsymbol{0}$ where $\boldsymbol{B} = \boldsymbol{I} - \boldsymbol{J}$. It will be convenient to also consider the transformed system $\bar{\boldsymbol{B}}\boldsymbol{y} = \boldsymbol{0}$, where

$$\bar{\boldsymbol{B}} = \boldsymbol{\Delta}^{-1}\boldsymbol{B} = \begin{pmatrix} \delta_1^{-1} & 1 & \cdots & 1 \\ 1 & \delta_2^{-1} & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & \delta_m^{-1} \end{pmatrix}$$

with $\boldsymbol{\Delta} = \mathrm{diag}\,(\delta_1, \ldots, \delta_m)$. Clearly, $\bar{\boldsymbol{B}}$ is symmetric, positive definite.

This linearized system will govern the behavior of the perturbation $\boldsymbol{y}$ when it is small, and so determine the behavior of the original nonlinear system within a local neighborhood of the equilibrium. The behavior of the sequential algorithm in a local neighborhood of the equilibrium can be approximated in terms of our linearized system of displacements with the relation

$$\boldsymbol{y}^{(n+1)} = (1-\omega)\boldsymbol{y}^{(n)} + \omega(\boldsymbol{J}^{\ell}\boldsymbol{y}^{(n+1)} + \boldsymbol{J}^{u}\boldsymbol{y}^{(n)}) = \omega\boldsymbol{J}^{\ell}\boldsymbol{y}^{(n+1)} + [(1-\omega)\boldsymbol{I} + \omega\boldsymbol{J}^{u}]\boldsymbol{y}^{(n)},$$

where $\boldsymbol{J}^{\ell}$ and $\boldsymbol{J}^{u}$ are respectively the lower and upper triangle matrices of $\boldsymbol{J}$. This approximating algorithm corresponds to the Gauss-Seidel matrix-iteration procedure $\boldsymbol{y}^{(n+1)} = \boldsymbol{G}_{\omega}\boldsymbol{y}^{(n)}$ for solving a linear system of equations (cf., e.g., [9]). Here the Gauss-Seidel matrix is given by $\boldsymbol{G}_{\omega} = (\boldsymbol{I} - \omega\boldsymbol{J}^{\ell})^{-1}[(1-\omega)\boldsymbol{I} + \omega\boldsymbol{J}^{u}]$, and arises from the splitting of the system matrix $\boldsymbol{B} = \boldsymbol{M}^{G}_{\omega} - \boldsymbol{N}^{G}_{\omega}$ with nonsingular $\boldsymbol{M}^{G}_{\omega} = \frac{1}{\omega}\boldsymbol{I} - \boldsymbol{J}^{\ell}$ and $\boldsymbol{N}^{G}_{\omega} = \frac{1-\omega}{\omega}\boldsymbol{I} + \boldsymbol{J}^{u}$.

6 Equivalence to the Power Problem

Here we remark that the net-benefit maximization model developed so far has an equivalent counterpart under the power criterion, a popular yet economically-controversial performance objective which has been extensively studied in the telecommunications literature (for surveys see [8], [5]). The power (throughput over average delay) criterion for user i takes the form

$$\max_{\lambda_i \in \mathcal{L}_i} P_i(\boldsymbol{\lambda}) = \max_{\lambda_i \in \mathcal{L}_i} \{\lambda_i^{\alpha_i}/W(\lambda)\}, \quad i = 1, \ldots, m,$$

where $W(\lambda)$ is the expected delay, and $P_i(\boldsymbol{\lambda})$ is the power objective for class i. Here, $\alpha_i > 0$ is a weighting factor that may be adjusted by user i to achieve different trade-off points between the average throughput and the expected time delay.

For an *M/M/1* queueing system the first-order, necessary and sufficient optimality conditions $\partial P_i/\partial \lambda_i = 0$ for an interior equilibrium provide the linear system of equations $\boldsymbol{P\lambda} = \boldsymbol{b}$ (cf., e.g., [10]), where

$$\boldsymbol{P} = \begin{pmatrix} 1 & \frac{\alpha_1}{\alpha_1+1} & \cdots & \frac{\alpha_1}{\alpha_1+1} \\ \frac{\alpha_2}{\alpha_2+1} & 1 & \cdots & \frac{\alpha_2}{\alpha_2+1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\alpha_m}{\alpha_m+1} & \frac{\alpha_m}{\alpha_m+1} & \cdots & 1 \end{pmatrix} \quad \text{and} \quad \boldsymbol{b}^{\mathrm{T}} = \left(\frac{\alpha_1}{\alpha_1+1}, \frac{\alpha_2}{\alpha_2+1}, \ldots, \frac{\alpha_m}{\alpha_m+1}\right)\mu.$$

Since the system is linear, no linearization in a local neighborhood about the equilibrium need take place. This affords the opportunity to establish global convergence results.

With the substitution $\delta_i = \alpha_i/(1+\alpha_i)$, the functional form of the net-benefit linearized system matrix $\boldsymbol{B}$ is the same as the system matrix $\boldsymbol{P}$ under the power criterion. Using the fact that $\delta_i \in (1/2, 1)$, we see that this assignment reduces the linearized version of the net-benefit problem to an instance of the power problem with $\alpha_i = \delta_i/(1-\delta_i) = (\delta_i^{-1} - 1)^{-1} > 1$. Since matrix-iterative algorithms (in particular the standard and relaxed Gauss-Seidel and Jacobi techniques) which solve these systems are simply splittings of the system matrix, the behavior of each system is the same. Thus, the global behavior of an *M/M/1* system operating under the power criterion with $\alpha_i > 1$ for all $i = 1, \ldots, m$ is equivalent to the local behavior of an *M/GI/1* system operating under the net-benefit criterion with linear delay costs and fixed rewards. We note here that any additional results derived for our net-benefit problem, particularly those developed for the relaxed algorithms, have an equivalent counterpart under the power criterion.

7 Convergence of the Relaxed Sequential Algorithm

In this section we examine the asymptotic behavior of the sequential algorithm. We first investigate convergence of the relaxed sequential algorithm. Since convergence will depend on the spectral radius of the relaxed Gauss-Seidel matrix $\boldsymbol{G}_\omega$, we appeal to a theorem of Kahan [4] (cf., [9], Theorem 3.5). which establishes $\rho(\boldsymbol{G}_\omega) \geq |1-\omega|$. Thus, the relaxed Gauss-Seidel iterative method diverges when $|1-\omega| \geq 1$, allowing us to restrict attention to the case $0 < \omega < 2$.

Theorem 1 *The relaxed sequential algorithm with $0 < \omega < 2$ converges asymptotically to the interior Nash equilibrium $\boldsymbol{\lambda}^*$ for all $m \geq 1$.*

Proof. We extend the proof of Theorem 1 in [10] to the relaxed algorithm. The relaxed Gauss-Seidel splitting $\bar{\boldsymbol{B}} = \bar{\boldsymbol{M}}^G_\omega - \bar{\boldsymbol{N}}^G_\omega$ with nonsingular $\bar{\boldsymbol{M}}^G_\omega = \boldsymbol{\Delta}^{-1}\boldsymbol{M}^G_\omega$ and $\bar{\boldsymbol{N}}^G_\omega = \boldsymbol{\Delta}^{-1}\boldsymbol{N}^G_\omega$ defines a corresponding relaxed Gauss-Seidel matrix

$$\begin{aligned}
\bar{\boldsymbol{G}}_\omega &= (\bar{\boldsymbol{M}}^G_\omega)^{-1}\bar{\boldsymbol{N}}^G_\omega = (\bar{\boldsymbol{M}}^G_\omega)^{-1}(\bar{\boldsymbol{M}}^G_\omega - \bar{\boldsymbol{B}}) \\
&= \boldsymbol{I} - (\bar{\boldsymbol{M}}^G_\omega)^{-1}\bar{\boldsymbol{B}} = \boldsymbol{I} - (\boldsymbol{\Delta}^{-1}\boldsymbol{M}^G_\omega)^{-1}\boldsymbol{\Delta}^{-1}\boldsymbol{B} = \boldsymbol{I} - (\boldsymbol{M}^G_\omega)^{-1}\boldsymbol{B} \\
&= (\boldsymbol{M}^G_\omega)^{-1}(\boldsymbol{M}^G_\omega - \boldsymbol{B}) = (\boldsymbol{M}^G_\omega)^{-1}\boldsymbol{N}^G_\omega = \boldsymbol{G}_\omega.
\end{aligned}$$

Since $\omega\bar{\boldsymbol{M}}^G_\omega = \boldsymbol{\Delta}^{-1}(\boldsymbol{I} - \omega\boldsymbol{J}^\ell)$ is nonsingular and $\bar{\boldsymbol{B}}$ is symmetric, positive definite for all ω, $\rho(\boldsymbol{G}_\omega) = \rho(\bar{\boldsymbol{G}}_\omega) < 1$ follows from Theorem 3.6, [9]. Thus, we have convergence for all $m \geq 1$. ■

8 The Case of Two User Classes

8.1 Acceleration of Convergence

We now examine the case of $m = 2$ user classes. In this case the 2×2 Jacobian matrix evaluated at the equilibrium is

$$\boldsymbol{G}_\omega = \begin{pmatrix} 1-\omega & -\omega\delta_1 \\ -\omega(1-\omega)\delta_2 & 1-\omega+\omega^2\delta_1\delta_2 \end{pmatrix}, \tag{12}$$

with Jacobian $\det \boldsymbol{G}_\omega = (1-\omega)^2$.

To optimize the rate of convergence of the relaxed sequential algorithm, we seek to find $\omega^* = \arg \min_{\omega \in (0,2)} \rho(\boldsymbol{G}_\omega)$. The characteristic equation in this 2-dimensional case is

$$v^2 - \operatorname{tr} \boldsymbol{G}_\omega v + \det \boldsymbol{G}_\omega = 0,$$

where $\operatorname{tr} \boldsymbol{G}_\omega = 2(1-\omega) + \omega^2\delta_1\delta_2$ and $\det \boldsymbol{G}_\omega = (1-\omega)^2$. The eigenvalues of $\boldsymbol{G}_\omega$ are given by

$$v_\pm(\omega) = \left(\operatorname{tr} \boldsymbol{G}_\omega \pm [(\operatorname{tr} \boldsymbol{G}_\omega)^2 - 4\det \boldsymbol{G}_\omega]^{1/2}\right)/2. \tag{13}$$

Since $\det \boldsymbol{G}_\omega > 0$, $(\operatorname{tr} \boldsymbol{G}_\omega)^2 - 4\det \boldsymbol{G}_\omega < (\operatorname{tr} \boldsymbol{G}_\omega)^2$. In the case of real eigenvalues this implies that $v_+(\omega) \geq v_-(\omega) > 0$. For the complex case the eigenvalues are complex conjugate. In either case, $\rho(\boldsymbol{G}_\omega) = |v_+(\omega)|$, where

$$v_+(\omega) = \left(2(1-\omega) + \omega^2\delta_1\delta_2 + \omega[\delta_1\delta_2]^{1/2}[4(1-\omega) + \omega^2\delta_1\delta_2]^{1/2}\right)/2.$$

The eigenvalues are real-valued if and only if $4(1-\omega)+\omega^2\delta_1\delta_2 \geq 0$. The expression $4(1-\omega)+\omega^2\delta_1\delta_2$ is equal to $\delta_1\delta_2 > 0$ at $\omega = 1$ and becomes $4(\delta_1\delta_2-1) < 0$ at $\omega = 2$. Thus, on the parameter space of interest, $\omega \in (0,2)$, the eigenvalues are real for $\omega \in (0, \omega_0]$ and complex for $\omega \in (\omega_0, 2)$, where $\omega_0 \in (1,2)$ is the smallest root of $4(1-\omega)+\omega^2\delta_1\delta_2 = 0$, namely,

$$\omega_0 = \frac{2(1-[1-\delta_1\delta_2]^{1/2})}{\delta_1\delta_2} \in (1,2).$$

We now proceed to establish that $\omega^* = \omega_0$ is fastest route to equilibrium for two user classes.

Theorem 2 *For $m = 2$ user classes, the rate of convergence of the relaxed sequential algorithm to the interior Nash equilibrium $\boldsymbol{\lambda}^*$ is maximized at $\omega^* = \omega_0$.*

Proof. To establish the result it suffices to show that $\rho(\boldsymbol{G}_\omega) = |v_+(\omega)|$ is monotonically decreasing on $\omega \in (0, \omega_0]$ and monotonically increasing on $\omega \in (\omega_0, 2)$. We consider the latter case first, that is, the complex case where $(\mathrm{tr}\, \boldsymbol{G}_\omega)^2 - 4 \det \boldsymbol{G}_\omega < 0$. In this case,

$$\rho^2(\boldsymbol{G}_\omega) = |v_\pm(\omega)|^2 = \left((\mathrm{tr}\, \boldsymbol{G}_\omega)^2 + 4 \det \boldsymbol{G}_\omega - (\mathrm{tr}\, \boldsymbol{G}_\omega)^2\right)/4 = \det \boldsymbol{G}_\omega = (1-\omega)^2, \quad (14)$$

which is increasing on $\omega \in (\omega_0, 2)$.

We now examine the real case where $4(1-\omega)+\omega^2\delta_1\delta_2 \geq 0$. To establish that $v_+(\omega)$ is decreasing on $\omega \in (0, \omega_0]$, we must establish that $(d/d\omega)v_+(\omega) < 0$, or equivalently,

$$(\omega\delta_1\delta_2 - 1)[\delta_1\delta_2\omega^2 - 4\omega + 4]^{1/2} + (\delta_1\delta_2)^{1/2}(\delta_1\delta_2\omega^2 - 3\omega + 2) < 0. \quad (15)$$

Note first that the expression $\delta_1\delta_2\omega^2 - 3\omega + 2$ is equal to $\delta_1\delta_2 - 1 < 0$ at $\omega = 1$ and becomes $4(\delta_1\delta_2 - 1) < 0$ at $\omega = 2$. This implies that second summand in (15) is negative for all ω on the interval $[1,2]$.

The sign of the first summand in (15) depends on the sign of $\omega\delta_1\delta_2 - 1$. Suppose first that $\omega\delta_1\delta_2 - 1 \geq 0$, i.e., $\omega \geq (\delta_1\delta_2)^{-1} > 1$. This implies that the first summand in (15) is positive and the second summand is negative so that (15) is satisfied if and only if the square of the first summand is less than square of the second, i.e.,

$$(\omega\delta_1\delta_2 - 1)^2[\delta_1\delta_2\omega^2 - 4\omega + 4] < \delta_1\delta_2(\delta_1\delta_2\omega^2 - 3\omega + 2)^2. \quad (16)$$

Expansion and cancelation of the terms in (16) yields an equivalent condition $4(1-\omega)(1-\delta_1\delta_2) < 0$, which is clearly satisfied.

Now suppose $\omega\delta_1\delta_2 - 1 < 0$. If $\omega < 1$ then the first summand in (15) is negative and it suffices to show that the square of the first summand exceeds the square of the second, i.e., $4(1-\omega)(1-\delta_1\delta_2) > 0$. Again, this condition is clearly satisfied. If $\omega \geq 1$ then both summands in (15) are negative as required. ■

8.2 A Hopf Bifurcation

We now examine the transition from a stable to an unstable equilibrium in the case of $m = 2$ user classes using the relaxed sequential algorithm. Mappings in $\Re^2$ often undergo a Hopf bifurcation in which stability passes from an attracting fixed point equilibrium to a two-dimensional attractor (often a circle) (cf., e.g., [3]).

Away from the equilibrium, the Jacobian matrix $\boldsymbol{G}_\omega$ in (12) has (using (7)) δ_i replaced by $1-(1/2)(\nu_i/\bar{\lambda}_i)^{1/2}$ for $i=1,2$. However, the Jacobian remains $(1-\omega)^2$, independent of the displacement from the equilibrium, $\boldsymbol{y}$. Therefore the map is a uniform contraction for $|1-\omega|<1$, area-preserving for $|1-\omega|=1$ and a uniform expansion for $|1-\omega|>1$.

At the point of transition to instability when $\omega = 2 > \omega_0$, the eigenvalues in (13) are complex conjugate and are given by $v_\pm = 2\delta_1\delta_2 - 1 \pm 2\delta_1\delta_2[1-\delta_1\delta_2]^{1/2}$. Rewriting these eigenvalues in polar form, $e^{\pm i\theta}$, with $\tan\theta = \frac{2\sqrt{\varsigma}}{1-\varsigma}$ and $\varsigma = \frac{1-\delta_1\delta_2}{\delta_1\delta_2} \in (0,3)$, we see that the eigenvalues are of unit modulus. As $\varsigma \uparrow 1$, $\tan\theta \to \infty$; i.e., $\theta \uparrow \pi/2$. As $\varsigma \downarrow 1$, $\tan\theta \to -\infty$; i.e., $\theta \downarrow \pi/2$. Thus, θ is increasing in ς, taking values in the range $(0,\pi/3)$. Clearly, $v_\pm^j \neq 1$ for $j=1,2,3$. It is possible that $v_\pm^4 = 1$, but only when $\varsigma = 1$. This leads us to establish the existence of a Hopf bifurcation in this setting when $\varsigma \neq 1$, i.e., $\delta_1\delta_2 \neq 1/2$.

Theorem 3 *Under the relaxed sequential algorithm with* $m=2$ *user classes there is a Hopf bifurcation of the interior Nash equilibrium* $\boldsymbol{\lambda}^*$ *at* $\omega = 2$ *provided* $\delta_1\delta_2 \neq 1/2$.

Proof. We have established that the eigenvalues $v_\pm$ are of unit modulus, and that when $\varsigma \neq 1$ the eigenvalues are not first, second, third or fourth roots of unity. It remains only to show that the eigenvalues cross the unit cycle with nonzero speed, viz.,

$$\frac{d}{d\omega}|v_\pm(\omega)|\bigg|_{\omega=2} = \frac{d}{d\omega}|1-\omega|\bigg|_{\omega=2} = 1 \neq 0,$$

where the first equality follows from (14). ∎

To illustrate this phenomenon, we consider an *M/M/1* queue with time units scaled in terms of the average service time $1/\mu = 1$. Let $h_i = 1$ be the delay cost rate for both user classes, and let the service rewards for classes 1 and 2 be $r_1 = 4$ and $r_2 = 8$, respectively. The corresponding cost/reward ratios are $\nu_1 = 1/4$ and $\nu_2 = 1/8$. Substitution into (11) yields "excess" capacity of $\nu = 1/3$ at the equilibrium $\lambda_1^* = 1/9$, $\lambda_2^* = 5/9$ given by (9). The Jacobi parameters $\delta_1 = 5/8$ and $\delta_2 = 13/16$ yield $\varsigma = 63/65$. Figure 1 illustrates that, at the Hopf bifurcation value $\omega = 2$, there is neither convergence nor divergence towards the equilibrium. As predicted by Theorem 3, a closed curve surrounding the equilibrium forms the domain of attraction for the map with $\omega = 2$. In general, this attracting set will depend on the starting point.

9 Conclusions

Using an *M/GI/1* model of a network link, a unique interior Nash equilibrium point among several classes of users was found. A relaxed sequential algorithm for obtaining the equilibrium was proposed and shown to converge in a neighborhood of the equilibrium for any number of user classes over an interval of the relaxation parameter. In the case of two user classes, relaxation was shown to accelerate convergence to equilibrium with fastest convergence provided by an overrelaxation. Additionally, in the transition from convergence to divergence, it was shown that there exists a Hopf bifurcation point at which the algorithm neither converges to nor diverges from equilibrium. Although these asymptotic results were developed for a network of users with an objective of net-benefit maximization, it was shown that these results translate directly to an *M/M/1* network of users seeking to maximize power.

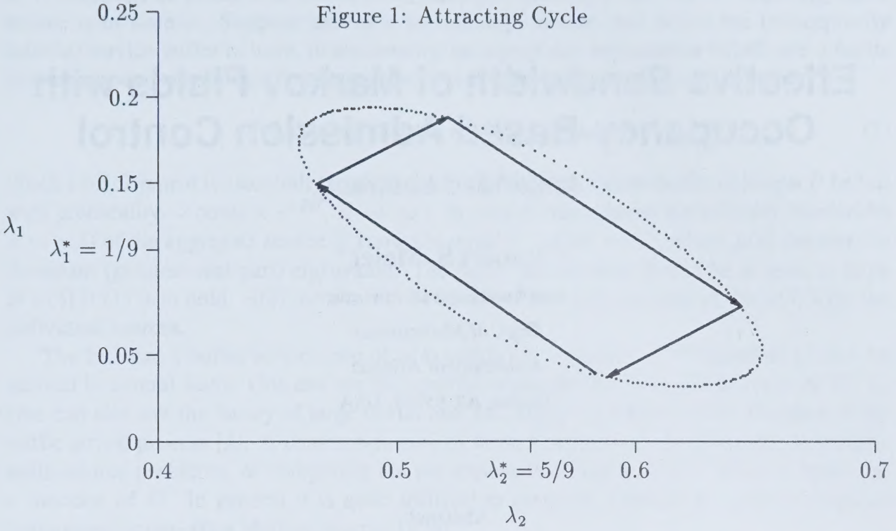


Figure 1: Attracting Cycle

References

[1] D. G. Bails and L. C. Peppers. *Business Fluctuations.* Prentice Hall, Englewood Cliffs, NJ, 2 edition, 1993.

[2] D. Gross and C. M. Harris. *Fundamentals of Queueing Theory.* John Wiley & Sons, New York, 2 edition, 1985.

[3] J. Guckenheimer and P. Holmes. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, volume 42 of *Applied Mathematical Sciences.* Springer-Verlag, New York, 1983.

[4] W. Kahan. *Gauss-Seidel methods of solving large systems of linear equations.* PhD thesis, University of Toronto, 1958.

[5] Y. A. Korilis and A. A. Lazar. Why is flow control hard: Optimality, fairness, partial and delayed information. Technical report, Columbia University, 1992.

[6] D. G. Luenberger. *Introduction to Dynamic Systems.* John Wiley & Sons, New York, 1979.

[7] C. M. Rump and S. Stidham, Jr. Asymptotic behavior of relaxed asynchronous flow-control algorithms for multiclass networks. Technical report, 1995. in preparation.

[8] S. Stidham, Jr. Optimal control of admission to a queueing system. *IEEE Transactions on Automatic Control*, AC-30(8):705–713, 1985.

[9] R. S. Varga. *Matrix Iterative Analysis.* Prentice-Hall, Englewood Cliffs, NJ, 1962.

[10] Z. Zhang and C. Douligeris. Convergence of synchronous and asynchronous algorithms in multiclass networks. In *Proceedings of the IEEE INFOCOM '91*, pages 939–943, Bal Harbor, FL, 1991.

Effective Bandwidth of Markov Fluids with Occupancy-Based Admission Control

(extended abstract)

Robert S. Maier*
rsm@math.arizona.edu
Dept. of Mathematics
University of Arizona
Tucson, AZ 85721, USA

Abstract

We consider the performance of a single-buffer statistical multiplexer, when the aggregate traffic source is modelled as a Markov-modulated fluid. An *occupancy-based* admission control policy would take into account the current buffer occupancy when deciding whether or not to admit traffic of any specified priority class. We show that the now standard effective bandwidth concept can be extended to nontrivial occupancy-based admission control policies. The effective bandwidth, which is the minimum service rate needed to ensure a specified asymptotic exponential falloff of the buffer occupancy probability density, turns out to be additive over homogeneous sources, though not over heterogeneous ones. We also show how the pre-exponential factor in the probability density asymptotics can be computed. Our asymptotic analysis uses the method of multiple scales, applied to the system of ordinary differential equations that determines the stationary buffer occupancy distribution.

1 Introduction

The effective bandwidth concept has proved its usefulness in the analysis of statistical multiplexing [1], where many traffic sources may be multiplexed into a single service buffer. Irrespective of whether the sources are modelled as Markov fluids, as in ATM-based networks, or point processes, an effective bandwidth may be associated to each source. This quantity depends on the service requirements, *e.g.*, the expected service delay, or the characteristics of the stationary buffer occupancy distribution. The buffer service rate should be at least as large as the sum of the effective bandwidths of the multiplexed sources, if the service requirements are to be met. This criterion in fact *defines* the effective bandwidth of many classes of Markov source [2].

Suppose for example that a single-buffer multiplexer is driven by independent fluid sources, and that the aggregate traffic arrival stream may be modelled as a Markov-modulated fluid with irreducible transition matrix M indexed by $s \in \Sigma$, the set of aggregate source states. Let

---

*Partially supported by the National Science Foundation under grants NCR-90-16211 and DMS-95-00792.

$\Lambda = \mathrm{diag}(\Lambda_{ss})$ be a matrix of arrival rates, Λ_{ss} being the rate of arrivals when the aggregate source is in state s. Suppose that as a service requirement, one wants the (conceptually infinite) service buffer to have, in stationarity, an asymptotic exponential falloff rate δ for its tail occupancy probability. That is, one requires that the buffer occupancy X satisfy

$$\mathsf{P}\{X > x\} \sim \text{const} \times e^{-\delta x}, \qquad x \to \infty. \tag{1}$$

(Such a requirement is essentially equivalent to requiring that a *finite* buffer of length B be full with probability $\sim$ const $\times\, e^{-\delta B}$, $B \to \infty$.) In such circumstances the effective bandwidth $\alpha = \alpha(\delta)$ of the aggregate source is known to equal $\delta^{-1}\mu(M + \delta\Lambda)$, where $\mu(\bullet)$ denotes the dominant (greatest-real-part) eigenvalue. The buffer service rate should be at least as large as $\alpha(\delta)$ if (1) is to hold. $\alpha(\delta)$ turns out to be additive: it equals the sum of the $\alpha(\delta)$'s for the individual sources.

The fact that a buffer service rate of $\alpha(\delta)$ yields the $\sim$ const $\times\, e^{-\delta x}$ falloff of (1) can be derived in several ways. One can use the spectral expansion technique of Mitra *et al.* [1, 3]. One can also use the theory of large deviations, *i.e.*, apply the Gärtner-Ellis Theorem to the traffic arrival process [2]. A common feature of these treatments is the difficulty, in realistic multi-source problems, of computing the pre-exponential factor in (1). What is 'const' as a function of δ? In general it is quite difficult to compute, even in the case of multiple homogeneous two-state Markov sources [3, 4].

The preceding discussion assumed that the traffic entering the multiplexer is not subject to any sort of occupancy-based admission control. As a consequence, the effective arrival rate matrix Λ is independent of the current buffer occupancy X. A more sophisticated stochastic model was considered by Elwalid and Mitra [5], in their analysis of an AT&T ATM switch. They analysed the performance of statistical multiplexing in ATM, with *loss priorities*. This means that the sources produce Markov fluids of various priority classes, and 'hard' admission control policies are applied. (When X/B rises above a certain class-dependent rejection level, no cells of that class are admitted.) As a consequence, the arrival rate matrix Λ may be a discontinuous, but piecewise constant, function of X/B. Elwalid and Mitra studied the effects of rejection level choice on the tradeoff between the loss of low-priority cells and the delay of high-priority cells. Their analysis was in terms of eigenvector expansions, and did not explicitly use the effective bandwidth concept. In fact they did not focus on the question of $B \to \infty$ asymptotics.

Motivated by the work of Elwalid and Mitra, in this paper we extend the effective bandwidth concept to quite general occupancy-based and priority-class-based admission control policies. This includes 'soft' rejection policies that reduce acceptance probabilities to zero *continuously*, rather than discontinuously. A very simple example is as follows. Suppose that the traffic sources (each modelled as a Markov fluid) generate traffic of two classes, a high-priority class and a low-priority class, and that the service buffer has a finite length B. High-priority traffic is always admitted (unless the buffer is full), but the probability of admission of low-priority traffic depends in some fixed way on the current buffer occupancy fraction X/B. For example, if $X/B > 1/2$ all low-priority traffic could be rejected; this would be a 'hard' policy of the type considered by Elwalid and Mitra. Alternatively, one could impose a 'soft,' or probabilistic, occupancy-based admission control policy: one could let the probability of admitting low-priority traffic decline linearly from 1 to 0 as X/B increases from $1/2$ to 1. The possibilities along this line are endless. It may well be advantageous to explore such control policies when studying the tradeoffs between cell loss probabilities and cell delay statistics.

We shall show that even for general occupancy-based admission control policies, for which each Λ_{ss} is allowed to be an arbitrary decreasing function of X/B, the effective bandwidth concept is still useful. The probability that arriving traffic will find the buffer full will always decline exponentially in B, in the $B \to \infty$ limit. In fact one can arrange any sufficiently rapid

$$\sim \text{const} \times e^{-\delta B}, \qquad B \to \infty \tag{2}$$

exponential falloff by choosing a sufficiently large service rate. This service rate must be at least as large as an 'effective bandwidth' for the aggregate traffic source. This bandwidth $\alpha = \alpha(\delta)$ is defined in a new way, which takes the occupancy-based admission control policy into account. The effective bandwidth turns out to be additive over *homogeneous* sources, but not over heterogeneous ones. This non-additivity occurs not only for soft policies, but also for the more conventional hard ones; this fact has not previously been remarked on. We shall show how for any aggregate source model plus nontrivial admission control policy, $\alpha = \alpha(\delta)$ may be computed numerically.

A byproduct of our analysis will be a increased understanding of the pre-exponential factor in (1). As noted, even for a trivial admission control policy (no consideration of current occupancy, only one priority class), the computation of the pre-exponential factor is difficult. But this factor is largely determined by an empty-buffer boundary condition [3]. We shall see that imposing a nontrivial admission control policy modifies the pre-exponential factor in (1), for any specified exponential falloff rate δ, in an easily computable way. The pre-exponential factor becomes x-dependent, and the x-dependence can be computed from the occupancy-dependence of the admission control policy.

2 An Asymptotic Analysis

Suppose as above that M signifies the transition matrix of the aggregate Markov source, and that $\Lambda = \text{diag}(\Lambda_{ii})$ is a matrix of arrival rates, indexed by source state. On account of the occupancy-based admission control policy, each Λ_{ii} is allowed to be influenced by the normalized buffer occupancy $Z \triangleq X/B$. This dependence will be specified by a decreasing function $\Lambda_{ii} = \Lambda_{ii}(z)$, $z \in [0,1]$. If the buffer, when nonempty, is drained at rate c, the diagonal matrix $D(z) \triangleq \Lambda(z) - cI$ specifies the rate of occupancy growth in each source state.

Let the stationary distribution of the multiplexing system be denoted by $\pi(x)$, where the vector $\pi = (\pi_i)_{i\in\Sigma}$ is defined by $\pi_i(x) \triangleq \mathsf{P}\{X \le x, \text{ state} = i\}$. We shall usually regard π as a function of $z \triangleq x/B$ rather than of x, *i.e.*, $\pi_i(z) \triangleq \mathsf{P}\{Z \le z, \text{ state} = i\}$. This vector function of the (normalized) occupancy must satisfy the stationary Kolmogorov equation [1]

$$\left[\frac{d}{dx}\pi(x)\right] D(x) = \pi(x)M. \tag{3}$$

Since the x-dependence of D is by assumption a dependence on z, it is best to write the Kolmogorov equation in terms of z. It becomes

$$\epsilon \left[\frac{d}{dz}\pi(z)\right] D(z) = \pi(z)M \tag{4}$$

where $\epsilon \triangleq B^{-1}$. For most nontrivial admission control policies, this system of coupled ordinary differential equations cannot be solved in closed form.

It is nonethless possible to extract information about $\epsilon \to 0$ (*i.e.*, $B \to \infty$) asymptotics of $\pi(z)$ at any fixed z by using the methods of singular perturbation theory; in particular, the *method of multiple scales*. We shall follow the approach of Anile *et al.* ([6], §2.3 and §3.2). Using their approach, one seeks a solution for $\pi(z)$ that is a linear combination of asymptotic series of the form

$$e^{-W(z)/\epsilon} \sum_{m=0}^{\infty} \epsilon^m \pi^{(m)}(z), \tag{5}$$

for certain functions $W = W(z)$ and $\pi^{(m)} = \pi^{(m)}(z), m = 0, 1, 2, \ldots$ The name of the method is justified because the exponential factor $e^{-W(z)/\epsilon}$ will vary on a $\Delta z = O(\epsilon)$ or $\Delta x = O(1)$ lengthscale, whereas the factors $\pi^{(m)}(z)$ will vary on a $\Delta z = O(1)$ or $\Delta x = O(B)$ lengthscale. Previous direct solutions of the Kolmogorov equation [1, 3] have included the exponential factor, but not the 'slowly varying' factors. In the absence of a 'soft' occupancy-based admission control policy, the additional slow variation is not present in the solution.

Substituting the series (5) into the Kolmogorov equation, and equating the coefficients of the various powers of ϵ on the two sides, yields a set of 'balance equations.' The first such equation is the so-called eikonal equation

$$\pi^{(0)}(z) H\left(z, W'(z)\right) = 0 \tag{6}$$

where the prime denotes differentiation with respect to z, and

$$H(z, p) \triangleq M + pD(z). \tag{7}$$

Since $\pi^{(0)}(z)$ is assumed to be nonzero for all z, it follows that for all z, $W'(z)$ must equal one of the values $p_j = p_j(z)$ for which the matrix $H(z, p)$ is singular. Determining the possible values p_j is a spectral problem, discussed at length by Elwalid and Mitra [1]. Suffice it to say that the values p_j with positive real part may be indexed as

$$0 < \operatorname{Re} p_1 \le \operatorname{Re} p_2 \le \cdots \tag{8}$$

and that normally, p_1 is real and $p_1 < \operatorname{Re} p_j$ for all $j > 1$. The value p_1 is the most important from the point of view of overflow asymptotics.

If the admission control policy is trivial (*i.e.*, is independent of z), then H is independent of z, and there is no need for any of the $m \ge 1$ terms in the series (5). The series reduces to

$$e^{-p_j z/\epsilon} \pi^{(0;j)}, \tag{9}$$

where $\pi^{(0;j)}$ is the left eigenvector of $H(p_j)$ with zero eigenvalue. Each such term will satisfy the Kolmogorov equation, but one must take a linear combination of such terms to satisfy the empty-buffer boundary condition [1, 3]

$$D_{ii} > 0 \implies \pi_i(0) = 0. \tag{10}$$

The full solution may be written as

$$\pi(z) = \sum_{j:\ \operatorname{Re} p_j > 0} a_j e^{-p_j z/\epsilon} \pi^{(0;j)} + w, \tag{11}$$

where w is a left eigenvector of H corresponding to $p = 0$, in fact the stationary probability vector of M. As $\epsilon = B^{-1} \to 0$, the stationary occupancy *density* $\pi' \triangleq d\pi/dz$ will have asymptotics

$$\pi'(z) \sim C\epsilon^{-1}e^{-p_1 z/\epsilon}\pi^{(0;1)} \tag{12}$$

since in the limit, the $j = 1$ term is dominant. The pre-exponential factor C equals $-a_1 p_1$.

The well known series (11) generalizes immediately to the case of a nontrivial admission control policy. If one keeps only the $m = 0$ (*i.e.*, $O(\epsilon^0)$) terms in (5), one will have

$$\pi(z) \sim \sum_{j:\ \mathrm{Re}\, p_j > 0} a_j e^{-W_j(z)/\epsilon}\pi^{(0;j)}(z) + w \tag{13}$$

as an *asymptotic* ($\epsilon = B^{-1} \to 0$) approximation to $\pi(z)$ at any fixed z. Here $W_j(z)$ must satisfy the condition that its derivative $W_j'(z)$ equal $p_j(z)$, the j'th solution of $\det H(z, p) = 0$. The coefficients $\{a_j\}$ are determined by the empty-buffer boundary condition, which becomes

$$D_{ii}(0) > 0 \implies \pi_i(0) = 0, \tag{14}$$

i.e., since we can choose each W_j to satisfy $W_j(0) = 0$,

$$D_{ii}(0) > 0 \implies \sum_{j:\ \mathrm{Re}\, p_j > 0} a_j \pi^{(0;j)}(z) + w = 0. \tag{15}$$

For nontrivial occupancy-based admission policies, we see that it is the policy that applies at $z = 0$ (*i.e.*, the *empty-buffer policy*) that determines the coefficients $\{a_j\}$.

To determine the z-dependence of the slowly varying factor in the asymptotic expression for π', we need to determine the normalization of the z-dependent eigenvector $\pi^{(0;1)}$. Up to this point we have used only the first of the balance equations, *i.e.*, the eikonal equation (6). As Anile *et al.* discuss, from the other balance equations one can deduce the following. Let $L^{(1)} = L^{(1)}(z)$ and $R^{(1)} = R^{(1)}(z)$ be the left and right eigenvectors of $H(z, p_1(z))$ with zero eigenvalue, normalized so that $L^{(1)}R^{(1)} = 1$. (The normalization is otherwise allowed to depend arbitrarily on z.) Then one can write

$$\pi^{(0;1)}(z) = \Pi^{(1)}(z)L^{(1)}(z), \tag{16}$$

in which the scalar function Π must satisfy a 'transport' equation (cf. [6], Ch. 3, eq. 2.14)

$$(L^{(1)}DR^{(1)})(z)\Pi^{(1)\prime}(z) + \left(L^{(1)\prime}DR^{(1)}\right)(z)\Pi^{(1)}(z) = 0. \tag{17}$$

If the dominant left and right eigenvectors $L^{(1)}$ and $R^{(1)}$ are computable at any fixed z, this transport equation can be integrated numerically from $z = 0$ to any greater value of z, to yield the correct normalizing factor $\Pi^{(1)}(z)$.

3 A Classical-Mechanical Restatement

We summarize the results of the last section. We have shown that in any stable Markov fluid model of a multiplexer with an occupancy-based admission policy, the stationary density vector $\pi' = \pi'(z)$ has $\epsilon = B^{-1} \to 0$ asymptotics

$$\pi'(z) \sim C\epsilon^{-1}e^{-W_1(z)/\epsilon}\Pi^{(1)}(z)L^{(1)}(z). \tag{18}$$

Here C is an overall normalization constant that follows from the empty-buffer boundary condition, and $W_1'(z)$ equals $p_1 = p_1(z)$, the dominant root of the polynomial equation $\det \boldsymbol{H}(z,p) = 0$. Also $\boldsymbol{L}^{(1)} = \boldsymbol{L}^{(1)}(z)$ is the left eigenvector of $\boldsymbol{H}(z,p_1)$, arbitrarily normalized, and the scalar function $\Pi^{(1)}$ satisfies the transport equation (17). The factor $\Pi^{(1)}(z)$ serves to compensate for the fact that the normalization of $\boldsymbol{L}^{(1)}(z)$ might have been incorrectly chosen.

There is a restatement of this result in the language of classical mechanics, which is both elegant and useful from a computational point of view. In the Appendix we review how, in the context of large deviation theory, a *classical Hamiltonian*, or energy function, naturally appears. For a multiplexer driven by a Markov-modulated fluid source, this function turns out to be of the form

$$H(z,p) = \mu\,[\boldsymbol{M} + p\boldsymbol{D}(z)]\,, \tag{19}$$

i.e., $H(z,p) = \mu\,[\boldsymbol{H}(z,p)]$, where $\mu\,[\bullet]$ denotes the dominant (greatest-real-part) eigenvalue. $H(z,p)$ is interpreted as the energy of a particle with momentum p, at the time it moves through position z. It is natural to restate the asymptotic results of the last section in terms of this Hamiltonian function. Since the dominant root p_1 has the property that $\boldsymbol{H}(z,p_1)$ has no eigenvalues in the right-half plane [1], we necessarily have an implicit *scalar* equation for $W_1 = W_1(z)$, namely

$$H(z, W_1'(z)) = 0. \tag{20}$$

In classical mechanics, this equation would be called a constant-energy constraint, or equivalently a *Hamilton-Jacobi equation* [7]. Its solution $W_1(z)$ would be called the *classical action* of the zero-energy classical trajectory beginning at 0, and extending to z. The derivative $W_1'(z)$ would be interpreted as the momentum $p = p(z)$ of the trajectory at the time it passes through z.

This is not just a formal analogy, because the asymptotic analysis of the last section can be efficiently restated in terms of Hamilton's equations of motion. Hamilton's equations, which are satisfied by any classical trajectory $t \mapsto (z,p)$, are of the form

$$\dot{z} = \frac{\partial H}{\partial p}(z,p) \tag{21}$$

$$\dot{p} = -\frac{\partial H}{\partial z}(z,p). \tag{22}$$

But it follows from (19), by standard first-order spectral perturbation theory, that

$$\frac{\partial H}{\partial p}(z,p_1) = (\boldsymbol{L}^{(1)}\boldsymbol{D}\boldsymbol{R}^{(1)})(z). \tag{23}$$

So the zero-energy trajectory giving rise to W_1 satisfies

$$\dot{z} = (\boldsymbol{L}^{(1)}\boldsymbol{D}\boldsymbol{R}^{(1)})(z). \tag{24}$$

This fact allows us to rewrite the transport equation (17) more compactly as

$$\dot{\Pi}^{(1)} + \left(\boldsymbol{L}^{(1)\prime}\boldsymbol{D}\boldsymbol{R}^{(1)}\right)(z(t))\,\Pi^{(1)} = 0. \tag{25}$$

The time derivative here is with respect to *transit time* along the zero-energy trajectory.

We conclude that in principle, any model of a single-buffer multiplexer driven by a Markov-modulated fluid source can be treated as follows. The Hamiltonian function $H = H(z,p)$ is defined by (19). For particular Markov fluid models (*e.g.*, models in which buffer is driven by homogeneous sources, each modelled as a Markov chain with only a few states) it may be possible to compute the Hamiltonian function in closed form. The momentum $p_1 = p_1(z)$ of the zero-energy trajectory would be computed numerically, by solving the zero-energy constraint equation $H(z, p_1(z)) = 0$. H is by examination convex in p, and at any z there will be only one nontrival (*i.e.*, nonzero) solution for p_1. The function $W_1(z)$ appearing in the asymptotic formula (18) simply equals the indefinite integral of p_1.

By using Hamilton's equation (21), we can compute $\dot{z}$ at any point z on the trajectory. In other words, we can work out the trajectory numerically, as a map $t \mapsto z$. The transport equation (25) would be integrated along the trajectory to yield the normalizing function $\Pi^{(1)} = \Pi^{(1)}(z)$ in (18), if the slowly varying factor is of interest.

The relevance of constant-energy classical trajectories in telecommunications models has been pointed out by Boel, Varaiya and Walrand [8] and the present author [9, 10]. Their usefulness in simplifying the equation for the pre-exponential normalizing function $\Pi^{(1)}$ is not, however, well known. We should stress that the zero-energy trajectory yielding W_1 has an immediate physical significance. It is the *most probable path to a buffer overflow*, in the $\epsilon \to 0$ ($B \to \infty$) limit. For models with trivial admission control policies this 'optimal path' will be a straight line. But if the momentum p_1 (and hence the velocity $\dot{z}$) depends on z, then the optimal path will not be straight. If the 'hard' control policies of Elwalid and Mitra [5] are used, it will be piecewise straight. If the probability of admission to the buffer, when the aggregate source is in any specified state, depends continuously on the buffer occupancy, then the optimal path will be continuously differentiable.

4 Discussion

We can now explain how to compute the effective bandwidth for any Markov fluid source, coupled with a nontrivial occupancy-based admission control policy. It follows from the $\epsilon \to 0$ asymptotic behavior of $\pi'(z)$, as given by (18), that

$$\mathsf{P}\{Z > z\} \sim \hat{C}(z) e^{-W_1(z)/\epsilon}, \tag{26}$$

where the pre-exponential factor $\hat{C}(z)$ equals $C\Pi^{(1)}(z)\left[L^{(1)}(z)\mathbf{1}\right]/W_1(z)$. That is,

$$\mathsf{P}\{X > zB\} \sim \hat{C}(z) e^{-BW_1(z)}, \qquad B \to \infty. \tag{27}$$

By setting $z = 1$, we see that the probability that an arriving cell will find the buffer full has an asymptotic $W_1(1)$ exponential falloff rate.

If we impose as a service requirement a condition that the exponential falloff rate be δ, the effective bandwidth $\alpha = \alpha(\delta)$ will by definition be the buffer service rate c for which $W_1(1) = \delta$. The numerical procedure discussed above can be employed to compute $W_1(1)$ for any fixed service rate. That is, the function $c \mapsto W_1(1)$ can be computed. The inverse function will be the effective bandwidth function $\delta \mapsto \alpha$. This inversion can be performed numerically.

We see that the results of Kesidis *et al.* [2] on effective bandwidth emerge as a special case. If the admission control policy is *trivial*, then $p \equiv p_1$, the nonzero solution of $H(p) = 0$ (there is no z-dependence in H). Since $W_1(z) = p_1 z$ in this case, $W_1(1)$ simply equals p_1. And because $H(p) = H_0(p) - cp$, where

$$H_0(p) = \mu\left[M + p\Lambda\right], \tag{28}$$

the fact that $H(p_1) = 0$ implies that $H_0(p_1) - cp = 0$, *i.e.*,

$$c = H_0(p_1)/p_1. \tag{29}$$

So the inverse of the function $c \mapsto W_1(1)$ [*i.e.*, of the function $c \mapsto p_1$] is simply $\delta \mapsto H_0(\delta)/\delta$. And

$$\alpha(\delta) = H_0(\delta)/\delta = \mu\left[M + \delta\Lambda\right]/\delta \tag{30}$$

will be the effective bandwidth function. If the admission control policy is *nontrivial*, there will be no such simplification. In general, to compute the function $c \mapsto W_1(1)$ and its inverse, the effective bandwidth function, the numerical technique of the last section must be employed.

It is easy to verify that if the multiplexer is driven by N *homogeneous* sources, irrespective of the admission control policy the effective bandwidth function $\alpha = \alpha(\delta)$ is multiplied by N. If the aggregate source comprises N identical sources, the aggregate source state space is an N-fold Cartesian product, and by examination the aggregate matrix $H_0^{[N]} = H_0^{[N]}(z, p)$, *i.e.*, $\mu\left[M^{[N]} + p\Lambda^{[N]}(z)\right]$, is an N-fold Kronecker sum [1]. Since the dominant eigenvalue of a Kronecker sum is the sum of the dominant eigenvalues of the individual matrices, we have that $H_0^{[N]}(z, p) = NH_0^{[1]}(z, p)$, in an obvious notation. The zero-energy constraint, which determines the momentum trajectory $p_1^{[N]} = p_1^{[N]}(z)$, becomes

$$NH_0^{[1]}(z, p_1^{[N]}(z)) - cp_1^{[N]}(z) = 0. \tag{31}$$

It follows that the function $p_1^{[N]} = p_1^{[N]}(z)$ and its anti-derivative $W_1^{[N]}$ will be unaffected as N is increased, so long as the service rate c is multiplied by N. This indicates that increasing N simply multiplies the effective bandwidth by N. This conclusion is unaffected by the functional form of $H_0^{[1]}(z, p)$, *i.e.*, by the choice of admission control policy.

So in general, effective bandwidth is additive over homogeneous sources. It is not difficult, however, to construct nontrivial occupancy-based admission control policies for which it fails to be additive over *heterogeneous* sources. The details of this, and also a discussion of the extent to which the pre-exponential factors depend on the number N of identical sources, will appear elsewhere.

Appendix: Large Deviation Theory

In this appendix we briefly review how the frequency of large excursions away from equilibrium of a random process (*e.g.*, a buffer occupancy process) may be efficiently quantified in terms of Hamiltonian and Lagrangian functions.

If s denotes time, let $X = X(s)$ be a queue occupancy process. The queueing model could be the above Markov fluid model, or a more conventional non-fluid model. If B is a large parameter such as a (finite) maximum queue length, define a scaled, or normalized process $Z = Z(t)$ by

$$Z \triangleq X/B \tag{32}$$

$$t \triangleq s/B. \tag{33}$$

This choice of scaling (space and time are scaled by the same factor) does not affect velocities, such as the mean downward drift rate in the occupancy that will be present if the queue is stable. We allow the arrival rate to depend on the current *normalized* occupancy Z.

Large deviation theory deals with the $B \to \infty$ asymptotics of the probabilities of events associated to such scaled processes as $Z = Z(t)$ [11]. For example, if $\tilde{Z} = \tilde{Z}(t)$ is some nontrivial 'reference path' and $\zeta > 0$ is small, the quantity

$$\mathsf{P}\left\{\left|Z(t) - \tilde{Z}(t)\right| < \zeta \text{ for all } t \in [0, T]\right\}, \tag{34}$$

which is the probability of closely tracking $\tilde{Z}$ over a specified time interval $[0, T]$, will normally decay exponentially as $B \to \infty$. This is because Z tracking $\tilde{Z}$ would correspond to the original process $X(s)$ tracking $B\tilde{Z}(s/B)$ over the time interval $0 \le s \le BT$. In the $B \to \infty$ limit this would represent a *large excursion*, of magnitude $O(B)$. The probability of such an excursion occurring would normally be exponentially suppressed as $B \to \infty$.

For many classes of scaled random processes, such as scaled buffer occupancy processes, the quantity (34) [for ζ small] will have $B \to \infty$ asymptotics

$$\exp\left[-B\mathcal{I}\left(\tilde{Z}(\bullet)\right) + o(B)\right] \tag{35}$$

where the 'action' $\mathcal{I}\left(\tilde{Z}(\bullet)\right)$, a functional of $\tilde{Z}(\bullet)$, is defined by

$$\mathcal{I}\left(\tilde{Z}(\bullet)\right) \triangleq \int_0^T L\left(\tilde{Z}(t), \dot{\tilde{Z}}(t)\right)\, dt. \tag{36}$$

Here $L = L(z, \dot{z})$ is a model-dependent Lagrangian function, and the $o(B)$ term in (35) typically converges as $B \to \infty$ to a nonzero constant. As in classical mechanics the Lagrangian is the Legendre transform of a Hamiltonian $H = H(z, p)$, where p is a nonphysical 'momentum' variable. That is,

$$L(z, \dot{z}) = \sup_{-\infty < p < \infty} \left[\dot{z}p - H(z, p)\right]. \tag{37}$$

It is often easier to compute H in closed form, than to compute L in closed form.

Example. If the buffer is drained at a constant rate c, and the arrival rate is independent of buffer occupancy, the form (35) is closely related to the asymptotic results provided by the Gärtner-Ellis Theorem. In this case H and L are independent of z. If $A(u)$ denotes the number of arrivals in a time interval of length u then [2]

$$H(p) = \left[\lim_{u \to \infty} u^{-1} \log \mathsf{E}\left(\exp pA(u)\right)\right] - cp. \tag{38}$$

So H is simply the asymptotic log moment generating function for $A(u) - cu$.

The following two special cases have been worked out [2].

- The arrival process is a Markov-modulated fluid with Markov transition matrix M and diagonal arrival rate matrix Λ. Then $H(p) = \mu[M + p\Lambda] - cp$, where $\mu[\bullet]$ denotes dominant eigenvalue. That is, $H(p) = \mu[M + pD]$.
- The arrival process is a Markov-modulated Poisson process, *i.e.*, traffic arrivals times form a Poisson point process with intensity Λ_{ss}, where s is the state of a Markov chain with transition matrix M. In this case $H(p) = \mu[M + (e^p - 1)\Lambda] - cp$.

Both of these examples can be extended to cover the case when the traffic arrival rates depend on the normalized buffer occupancy. That is, the rate matrix Λ may depend on z. As a consequence, H will become a function of z as well as p. This is the case analysed analytically, rather than probabilistically, in Sections 2 and 3.

References

[1] A. I. Elwalid and D. Mitra, "Effective bandwidth of general Markovian traffic sources and admission control of high speed networks," *IEEE/ACM Trans. Networking*, vol. 1, no. 3, pp. 329–343, 1993.

[2] G. Kesidis, J. Walrand, and C.-S. Chang, "Effective bandwidths for multiclass Markov fluids and other ATM sources," *IEEE/ACM Trans. Networking*, vol. 1, no. 4, pp. 424–428, 1993.

[3] D. Anick, D. Mitra, and M. M. Sondhi, "Stochastic theory of a data-handling system with multiple sources," *Bell System Tech. J.*, vol. 61, no. 8, pp. 1871–1894, 1982.

[4] R. Guérin, H. Ahmadi, and M. Naghshineh, "Equivalent capacity and its application to bandwidth allocation in high-speed networks," *IEEE Trans. Sel. Areas Comm.*, vol. 9, no. 7, pp. 968–981, 1991.

[5] A. I. Elwalid and D. Mitra, "Fluid models for the analysis and design of statistical multiplexing with loss priorities on multiple classes of bursty traffic." In *IEEE Infocom '92*, volume 3, pp. 415–425.

[6] A. M. Anile, J. K. Hunter, P. Pantano, and G. Russo, *Ray Methods for Nonlinear Waves in Fluids and Plasmas*. New York: Longman, 1993.

[7] H. Goldstein, *Classical Mechanics*. Reading, Massachusetts: Addison-Wesley, second ed., 1980.

[8] R. Boel, P. B. Varaiya, and J. Walrand, "Averaged dynamics and large deviations theory for the analysis and synthesis of communication network protocols," in *Queuing, Performance and Control in ATM* (J. W. Cohen and C. D. Pack, eds.), pp. 225–230, New York/Amsterdam: Elsevier, 1991. Proceedings of the Thirteenth International Teletraffic Congress, Copenhagen, June 1991.

[9] R. S. Maier, "The instability times of multiaccess broadcast channels," in *Proceedings of the 28th Allerton Conference on Communication, Control and Computing*, (Monticello, Illinois), pp. 139–148, Oct 1990.

[10] R. S. Maier, "Large fluctuations in stochastically perturbed nonlinear systems: Applications in computing," in *1992 Lectures on Complex Systems* (L. Nadel and D. L. Stein, eds.), pp. 501–517, New York: Addison-Wesley, 1993. Proceedings of the 1992 Complex Systems Summer School (Santa Fé, June 1992).

[11] S. R. S. Varadhan, *Large Deviations and Applications*. Philadelphia: Society for Industrial and Applied Mathematics, 1984.

Spatial and Temporal Precoding and Equalization Using Adaptive Array Antenna for Mobile Radio Communications

Ryuji Kohno

Div. of Elec. & Comp. Eng., Faculty of Eng., Yokohama National University
156 Tokiwadai, Hodogaya, Yokohama, 240 Japan
Email: kohno@kohnolab.dnj.ynu.ac.jp

Abstract An adaptive array antenna is considered as a filter in spatial and temporal domains for radio communications, so that the communication theory can be generalized into spatial and temporal processing. From an information theoretical viewpoint, this paper describes spatial and temporal signaling, equalization and detection based on an adaptive array antenna. In particular, spataily and temporally joint equalization in both a transmitter and a receiver is proposed as an entire optimum system. Such spatial and temporal processing can achieve drastic improvement of the theoretical performance and must be practical countermeasures for multipath fading in mobile radio communications.

I. ADAPTIVE ARRAY ANTENNA

An adaptive array antenna is an antenna that controls its own pattern by means of feedback control while the antenna operates. An adaptive tapped delay line (TDL) array antenna in Fig.1, which has TDL or digital filter in each antenna element, also control their own frequency response. The pattern of an array is easily controlled by adjusting the amplitude and phase of the signal from each element before combining the signals.

When the input signal to the TDL array antenna is $x(t)$, the antenna output is represented by

$$y(t) = \sum_{n=0}^{N-1} \sum_{m=0}^{M} x(t - mT_0) w_{n,m} \exp(-jn\varphi), \quad (1)$$

where T_0 is the delay between adjacent taps, $w_{n,m}$ is the mth complex tap coefficient of the nth, N is the number of elements and M is the number of taps. φ is the phase difference between the received signal at adjacent antenna elements in a line array is given by

$$\varphi = \frac{2\pi L \sin\theta}{\lambda}, \quad (2)$$

where λ, L and θ are the wave-length of an input signal, the distance between adjacent elements and the arrival angle of the received signal or direction of arrival(DOA), respectively. The antenna transfer function in both spatial and temporal domains is written by

$$H(\omega, \theta) = \sum_{m=0}^{M} \exp(-jm\omega T_0) \sum_{n=0}^{N-1} w_{n,m} \exp(-jn\varphi) \quad (3)$$

Equation (3) represents the antenna pattern when ω is a constant, while it represents the frequency response when θ is a constant.

Therefore, the adaptive TDL array antenna can be employed as a tool of signaling, equalization and detection in spatial and temporal domains.

II. SPATIAL AND TEMPORAL CHANNEL MODEL

A multipath fading channel such as a mobile radio channel is modeled in which a transmitted signal from one signal source arrives at the

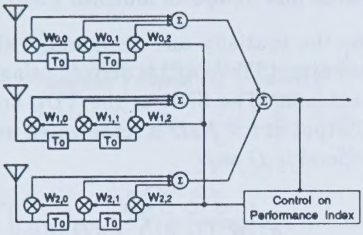


Fig. 1. An Adaptive TDL Array Antenna

receiver with different angles and delays. The received signal is represented by using time t and arrival angle θ.

Each propagation path in a channel is defined by its impulse response at the arrival angle θ of the received signal. The impulse response of the kth path $h_k(t)$ with the arrival angle θ_k, $(k = 1, 2, \cdots, K)$ is represented by

$$h_k(t) = \sum_{i=1}^{I_k} g_{k,i}\delta(t-\tau_{k,i})\exp(j\psi_{k,i}), \qquad (4)$$

where $g_{k,i}, \tau_{k,i}$ and $\psi_{k,i}$ denote path amplitude, path delay and path phase of the ith delayed signal through the kth path, respectively. I_k is the number of delayed signals in the kth path. $\delta(t)$ is the Dirac delta function.

An equivalent complex baseband signal of received signal $R_n(t, \theta_k)$ in the nth antenna element is written by

$$R_n(t,\theta_k) = \sum_{i=1}^{I_k} g_{k,i} S_b(t-\tau_{k,i}) \times \exp\left(-j2\pi nL\frac{\sin\theta_k}{\lambda}\right)\exp(j\phi_{k,i}) \qquad (5)$$

where $S_b(t)$ is the complex baseband transmitted signal and $\phi_{k,i}$ is the net phase offset.

III. SPATIAL AND TEMPORAL EQUALIZATION

A Spatial and Temporal Nyquest Theorem

The Nyquist criterion in both spatial and temporal domains can be derived from Eqs.(1) and (5). The array output $y(t)$ can be replaced by $y(t, \Theta)$ because the antenna output depends on time t and arrival angle set $\Theta = (\theta_1, \theta_2, \cdots, \theta_K)$. The sampled antenna output is rewritten as

$$y(lT_d;\Theta) = \sum_{k=1}^{K} p(t,\theta_k), \qquad (6)$$

where $p(t, \theta_k)$ is defined as

$$p(t,\theta_k) = \sum_{n=0}^{N-1}\sum_{m=0}^{M}\sum_{i=1}^{I_k} w_{n,m} g_{k,i}\exp(j\phi_{k,i}) \exp\left(-j2\pi nL\frac{\sin\theta_k}{\lambda}\right)S_b(t-\tau_{k,i}-mT) \qquad (7)$$

Assumed that θ_1 represents the desired arrival angle. If $p(t, \theta)$ equals to the symbol s_l at $t = lT_d$ and $\theta = \theta_1$, and equals to zero elsewhere, ISI must be zero. This condition is named the generalized Nyquist criterion in both spatial and temporal domains. Then, the criterion is represented by

$$p(lT_d,\theta) = s_l\delta'(t-lT_d, \theta-\theta_1), \qquad (8)$$

where $\delta'(t, \theta)$ is the two dimensional Dirac delta function and s_l represents the transmitted symbol at $t = lT_d$. This includes the usual Nyquist criterion when $p(t, \theta_k)$ is a function of time only.

B Adaptive S & T Equalization for Reducing ISI

Several criteria for spatial and temporal equalization such as ZF (Zero Forcing) and MSE (Mean Square Error), are available to update the weights and tap coefficients. The ZF criterion satisfies the generalized Nyquist criterion in case of noise free if there are infinite number of taps and elements. Since finite number of taps and elements is available in practical noisy multipath channels, there may be some equalization errors in adaptive equalization based on temporal updating algorithms. If the permissible equalization error is given, we can obtain several combinations of taps and elements which achieved the same equalization error. Therefore, the number of antenna elements can be reduced by increasing the number of taps when the difference in arrival angles is large.

A temporal updating algorithm of adaptive array antennas, such as LMS and Applebaum arrays, beamforms to track the desired signal and to suppress interference by nulls so as to maximize array output signal-to-noise ratio (SNR). Applebaum array is also useful when the arrival angle of the desired signal is known in advance. LMS array doesn't require any knowledge for the arrival angle of the desired signal. As long as reference signal correlated with the desired signal can be obtained, the array pattern will adaptively track the desired signal so as to maximize SNR. However, it is difficult to obtain a reliable reference signal in time-varying channels such as a mobile radio channel.

We have proposed and investigated an algorithm for controlling weights of antenna elements which are derived from spatial spectrum of spatially sampled signals by array of antenna elements. The arrival angles can be estimated from spatial frequency spectrum, which can be obtained by DFT or MEM for spatially sampled signals. The weight coefficients are updated by Wiener solution derived from the estimated spatial spectrum. Moreover, MUSIC algorithm estimates arrival angles in noise subspace which is defined by eigen vectors of covariance matrix of spatially sampled signals, while DFT and MEM do it in signal subspace. MUSIC has better estimation performance than MEM if noise subspace is larger for uncorrelated signals than signal subspace.

IV. SPATIAL AND TEMPORAL OPTIMUM RECEIVER

In the previous section, spatial and temporal equalization whose purpose is to reduce ISI due to multipath in a channel has been discussed. Viterbi equalization whose purpose is to achieve maximum likelihood sequence estimation utilizing ISI can be also generalized in spatial and temporal domains if an array antenna is employed. The tandem structure of a matched filter(MF) and a maximum likelihood sequence estimator(MLSE) or Viterbi Detector(VD) has been considered to be an optimum receiver in the presence of ISI and AWGN. The optimum receiver is generalized into spatial and temporal domains in this section.

A Spatial and Temporal Matched Filter

First, the spatially and temporally whitened matched filter (ST-WMF) is derived using a TDL array antenna. The SNR at the TDL array antenna output at $t = K_0 D$ is represented using the delay operator D as

$$\mathrm{SNR} = \frac{\sigma_s^2 \left| \mathbf{w}^T(D,\theta) \sum_{l=0}^{L-1} h_l(D)\mathbf{q}(\theta_l) \right|^2}{\sigma^2 \left|\mathbf{w}(D,\theta)\right|^2}, \tag{9}$$

where $h_l(D)$, $\mathbf{w}(D,\theta)$ and σ_s^2 denote the impulse response of th lth path, N-dimensional impulse response of the array at the arrival angle θ, and the variance of the input sequence $x(D)$, respectively. From Schwarz's inequality, the optimal weight vector $\mathbf{w}(D,\theta)$ (N-dimensional) so as to maximize the SNR at the TDL array antenna output is given by the time inversion $h_l(D^{-1})D^{K_0}$ ($l = 0, 1, \cdots, L-1$) of the impulse response and the directivity information ϕ as

$$\mathbf{w}(D,\theta) = \sum_{l=0}^{L-1} h_l(D^{-1})D^{K_0}\phi, \tag{10}$$

where K_0 satisfied the duration time $max_{l=0,1,\cdots,L-1}\{K_l\} \leq K_0$, the time inversion of the impulse response denotes a temporal WMF(T-WMF), and a directivity information denotes a spatial WMF(S-WMF) represented by $\phi = [w_0, w_1, \cdots, w_{N-1}]^T$. A S-WMF for the lth path is realized by the complex conjugated operation of received phase difference $\exp(jn\pi \sin\theta_l)$ due to the arrival angle. Therefore, a S-WMF coefficient w_n in the nth antenna element is

$$w_n = \sum_{l=0}^{L-1} \exp(jn\pi \sin\theta_l). \tag{11}$$

This is the generalization of the WMF in both spatial and temporal domains.

B Spatial and Temporal Optimum Receiver

Fig.2 shows a VD, which is connected to a ST-WMF built using a TDL array antenna. We call it a receiver with ST-WMF and VD or a spatial and temporal optimum receiver. As a special case, a receiver with a S-WMF & VD and

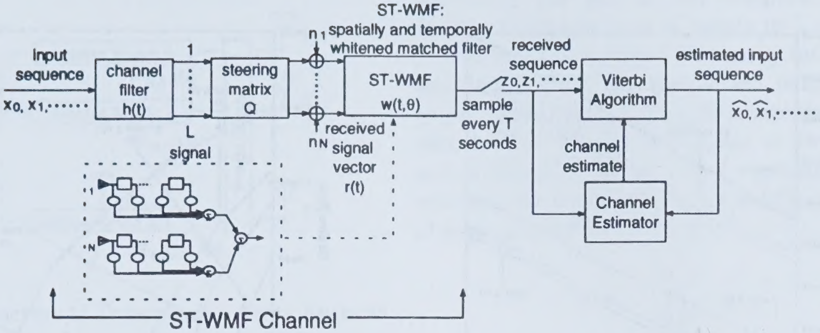

ST-WMF:
spatially and temporally
whitened matched filter
input
sequence
$x_0, x_1, \cdots\cdots$
channel
filter
h(t)
1
L
steering
matrix
Q
n_1
n_N
ST-WMF
w(t,θ)
received
signal
vector
r(t)
signal
1
N
sample
every T
seconds
received
sequence
$z_0, z_1, \cdots\cdots$
Viterbi
Algorithm
estimated input
sequence
$\widehat{x_0}, \widehat{x_1}, \cdots\cdots$
channel
estimate
Channel
Estimator
ST-WMF Channel

Fig. 2. Spatail and Temporal Optimum Receiver

that with a T-WMF & VD are included. The detection algorithm in the proposed receiver is described as follows. (i) Each antenna element receives signals. (ii)Received signals in each antenna element are filtered by a ST-WMF, which is matched to the transmission channel impulse response. (iii)The maximum likelihood sequence is estimated for the ST-WMF output. The symbol error probability $P(e)$ of the proposed optimal receiver in the spatial and temporal domains is bounded by

$$P(e) \leq \alpha Q(d_{min}/2\sigma), \qquad (12)$$

where d_{min} is the minimum Euclidean distance, and α is a small constant. $Q(\cdot)$ is the error function.

Since ISI is taken into account, the transmission rate $\mathcal{R}_{st}$ is derived as

$$\mathcal{R}_{st} = W \log \frac{\sigma_s^2 \sum_{k=0}^{2K-1} |g_k|^2 + \sigma^2}{\sigma^2}, \qquad (13)$$

where the input has variance σ_s^2. Fig.3 shows the BER and the transmission rate of the proposed ST-WMF and VD receiver in comparison with other receivers.

C Spatial and Temporal Optimum Multiuser Receiver for CDMA

A direct-sequence CDMA(DS/CDMA) mobile radio communication channel is modeled as a channel with both ISI due to multipath and co-channel interference(CCI) due to the correlation between spreading sequences of simultaneously accessing users. The optimum multiuser receiver for DS/CDMA detects every user's data in a sense of MLSE by utilizing CCI as redundant information which multiple accessing users share. By using an adaptive TDL array antenna, we derive such spatial and temporal optimum multiuser receiver that MLSE for every user's data can be achieved with both CCI and ISI in DS/CDMA.

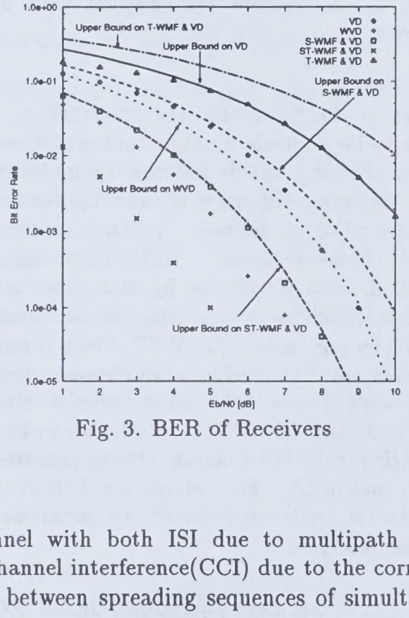


Fig. 3. BER of Receivers

The receiver has an extended structure in Fig.2 so that correlators for every user are located in

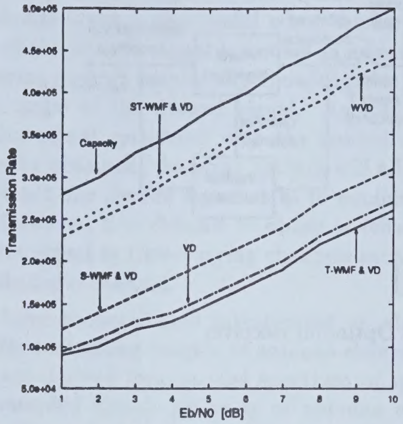


Fig. 4. Acheivable Transmission Rate of Receivers

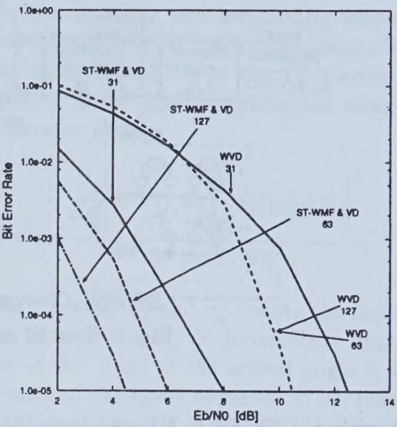


Fig. 5. BER of Multiuser Receivers

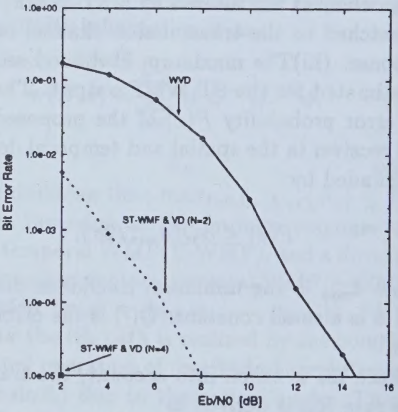


Fig. 6. BER of Multiuser Receivers

front of the ST-WMF, the ST-WMF is modified to be multiple input/output structure with cross coupling and is followed by multiple VD. The detection algorithm in the proposed receiver is described as follows. (i)Each antenna element receives signals. (ii)Received signals in each element are filtered by each user's correlator. (iii)Each user's correlator output vector is filtered by each user's ST-WMF, which is matched to each user's channel impulse response. (iv)Each user's maximum likelihood sequence is estimated for each user's ST-WMF output, where the path metric is calculated taking into consideration the influence of CCI. Fig.5 shows the BER of the ST optimum multiuser receiver in comparison with other receivers.

V. SPATIAL AND TEMPORAL JOINT EQUALIZER IN TRANSMITTER AND RECEIVER

A Signal Configuration for Transmitter and Receiver

The input sequence is defined using delay operator D (D-transform) as

$$x(D) = x_0 + x_1 D + x_2 D^2 + \cdots . \quad (14)$$

It is assumed that the signals can be transmitted for K directions. Figure 7 shows the transmission array antenna. The number of antenna weight sets is K and these antenna weight sets can be transmitted for K direction. The transmission array antenna in temporal part are characterized by the vector of finite impulse response $W_{t,i}(D)$ (i=0,1,$\cdots$,K−1) (M-dimensional) of length M_i symbol interval; i.e., M_i is the smallest integer such that $W_{t,i}(t)$=0 for $t \geq M_i T$. Where M donates the number of transmission antenna elements. The transmission vector is defined by

$$W_t(D) = \left[\begin{array}{cccc} W_{t,0}(D) & W_{t,1}(D) & \cdots & W_{t,K-1}(D) \end{array} \right] \quad (15)$$

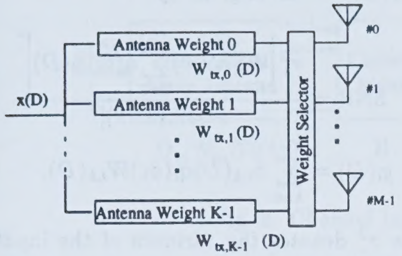


Fig. 7. Structure of Transmission Array Antenna

Then, this vector is changed to K-dimensional vector by transmission steering converter in spatial domain.

$$Q_t = \begin{bmatrix} q_t(\phi_0) & 0 & \cdots & 0 \\ 0 & q_t(\phi_1) & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & q_t(\phi_{K-1}) \end{bmatrix} \qquad (16)$$

Let the steering vector for bearing ϕ be $q_t(\phi)$ (K-dimensional) given by

$$q_t(\phi) = \begin{bmatrix} 1 & e^{-j\pi \sin\phi} & \cdots & e^{-j(M-1)\pi \sin\phi} \end{bmatrix}. \qquad (17)$$

The transmitted sequence is represented by the convolution of the input sequence $x(D)$ and the transmission array antenna weight $Q_tW_t(D)$ as

$$\begin{aligned} X(D) &= (Q_tW_t(D))^T x(D) \qquad (18) \\ &= \begin{bmatrix} X_0(D) & X_1(D) & \cdots & X_{K-1}(D) \end{bmatrix}^T. \end{aligned}$$

It is assumed that the channel for each arrival angle θ_l at the receiver array antenna is characterized by a vector

$$h_l(D) = \begin{bmatrix} h_{l,0}(D) & h_{l,1}(D) & \cdots & h_{l,K-1}(D) \end{bmatrix} \qquad (19)$$

where l=0,1,$\cdots$,$L-1$ and L denotes the total number of paths. We defined the path as its past history due to bearing. The l^{th} signal sequence $s_l(D)$ is represented by the transmitted sequence vector $X(D)$ and the l^{th} channel vector $h_l(D)$ as

$$\begin{aligned} s_l(D) &= h_l(D)X(D) = \sum_{k=0}^{K-1} h_{l,k}(D)X_k(D) \\ &= \sum_{k=0}^{K-1} h_{l,k}(D)q_t(\phi_k)W_{t,k}(D)x(D). \qquad (20) \end{aligned}$$

Therefore, the spatial and temporal channel change the signals from K inputs to L outputs.

Since a receiver using a TDL array antenna can sample signals in the spatial and temporal domains, each transmission path is represented not only by time but also by bearing or direction of arrival (DOA). If the receiver is a omnidirectional antenna, the signal sequence is simply composed of sequences through all paths as

$$s(D) = \sum_{l=0}^{L-1} s_l(D). \qquad (21)$$

It is assumed that

1. A uniform linear array (ULA) is employed in the receiver.
2. All incoming signals are plane waves.
3. Bearing is defined as the angle from broadside.

Let the reception steering vector for bearing θ be $q_r(\theta)$ ($N-$dimensional) given by

$$q_r(\theta) = \begin{bmatrix} 1 & e^{-j\pi \sin\theta} & \cdots & e^{-j(N-1)\pi \sin\theta} \end{bmatrix}, \qquad (22)$$

where N denotes the number of antenna elements.

When the signals are received by the array antenna which has N antenna elements, the signal sequence vector $\mathbf{r}_l(D, \theta_l)$ through the l^{th} path (N-dimensional) is given by D and the bearing θ_l as

$$\mathbf{r}_l(D, \theta_l) = q_r(\theta_l)s_l(D), \qquad (23)$$

where θ_l denotes the bearing of the l^{th} path. Then the received signal sequence vector $\mathbf{r}(D, \theta)$ actually is composed of signal sequence vectors through each path and a white Gaussian noise vector of zero mean and variance σ^2 as

$$\mathbf{r}(D, \theta) = \sum_{l=0}^{L-1} \mathbf{r}_l(D, \theta_l) + \mathbf{n}_0. \qquad (24)$$

In this section, it is assumed that the received signal via the first path is the desired one. Therefore, the received signal sequence vector $\mathbf{r}(D, \theta)$ is represented by the sum of the desired signal sequence vector $\mathbf{s}_0(D, \theta_0)$, the interfering signals sequence vectors $\mathbf{r}_l(D,\theta_l)$ $(l \neq 0)$ and the noise sequence vector $\mathbf{n}_0$ as

$$\mathbf{r}(D) = \mathbf{r}_0(D,\theta_0) + \sum_{l=1}^{L-1} \mathbf{r}_l(D,\theta_l) + \mathbf{n}_0. \tag{25}$$

Let the array impulse response vector be $W_r(D,\theta)$ ($N-$dimensional). The output sequence $z(D)$ is given by

$$z(D) = W_r^T(D,\theta)\mathbf{r}(D,\theta). \tag{26}$$

The channel model is represented by Fig. 8. The matrix H, Q_r and vector W_r are represented by

$$H = \begin{bmatrix} h_{0,0}(D) & h_{0,1}(D) & \cdots & h_{0,K-1}(D) \\ h_{1,0}(D) & h_{1,1}(D) & \cdots & h_{1,K-1}(D) \\ \vdots & \vdots & \vdots & \vdots \\ h_{L-1,0}(D) & h_{L-1,1}(D) & \cdots & h_{L-1,K-1}(D) \end{bmatrix}, \tag{27}$$

$$Q_r = \begin{bmatrix} q_r(\theta_0) & 0 & \cdots & 0 \\ 0 & q_r(\theta_1) & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & q_r(\theta_{L-1}) \end{bmatrix}, \tag{28}$$

$$W_r = \begin{bmatrix} W_{r,0}(D) & W_{r,1}(D) & \cdots & W_{r,N-1}(D) \end{bmatrix}. \tag{29}$$

B Derivation of the ST-WMF

If the noise spectral density is large, the conventional Viterbi decoder (VD) may produce an estimate that is different from the input sequence due to the influence of noise. To improve the performance, a whitened matched filter (WMF) is considered. A usual WMF, i.e., temporally WMF (T-WMF), maximizes the SNR at the filter output by compensating for time difference or ISI in the received signals after they are merged in a tapped delay line (TDL) array antenna. However, there are also phase differences between the received signals at the input of the TDL array antenna. That is, a TDL array antenna can sample signals in both spatial and temporal domains. Therefore, in this section we design a spatially and temporally WMF (ST-WMF) using a TDL array antenna. The derivation of a ST-WMF is as follows. The SNR at the TDL array antenna output at $t = \lambda D$ is given by

$$\mathrm{SNR} = \frac{\sigma_s^2 \left| W_r^T(D,\theta) \sum_{l=0}^{L-1} \mathbf{q}(\theta_l) g_l(D) \right|^2}{\sigma^2 \left| W_r(D,\theta) \right|^2} \tag{30}$$

$$g_l(D) = \sum_{k=0}^{K-1} h_{l,k}(D)\mathbf{q}_t(\phi_k)W_{t,k}(D), \tag{31}$$

where σ_s^2 denotes the variance of the input sequence $x(D)$. From Schwarz's inequality, the optimal weight vector $W_r(D,\theta)$ (N-dimensional) so as to maximize the SNR at the TDL array antenna output is given by the time inversion and conjugate $g_l^*(D^{-1})D^\lambda$ $(l = 0, 1, \cdots, L-1)$ of the impulse response and the directivity information $\mathbf{\Phi}$ as

$$w_r(D,\theta) = \sum_{l=0}^{L-1} g_l^*(D^{-1})D^\lambda \mathbf{\Phi}, \tag{32}$$

where λ satisfied the sum of the maximum delay of the channel and the delay of the transmission filter and * donates the complex conjugate. The time inversion and conjugate of the impulse response denotes a T-WMF, and a directivity information denotes a spatially WMF (S-WMF) represented by

$$\mathbf{\Phi} = [w_0, w_1, \cdots, w_{N-1}]^T. \tag{33}$$

A S-WMF for the l^{th} path is realized by the complex conjugate operation of received phase difference $\exp(jn\pi\sin\theta_l)$ due to bearing. Therefore, a S-WMF coefficient w_n in the n^{th} antenna element is

$$w_n = \sum_{l=0}^{L-1} \exp(jn\pi\sin\theta_l). \tag{34}$$

This is the generalization of the WMF in both spatial and temporal domains. A T-WMF and a S-WMF are considered to be special kinds of the above mentioned ST-WMF.

The temporally optimal weight vector $W_T(D)$ is given by making the directivity information $w_n = 1$ $(n = 0, 1, \cdots, N-1)$ in Eq. (32). That is

$$W_T(D) = \sum_{l=0}^{L-1} g_l^*(D^{-1})D^\lambda. \tag{35}$$

While, the spatially optimal weight vector $W_S(\theta)$ is consisted of only the directivity information $\mathbf{\Phi}$ so as to beamform to track the signal

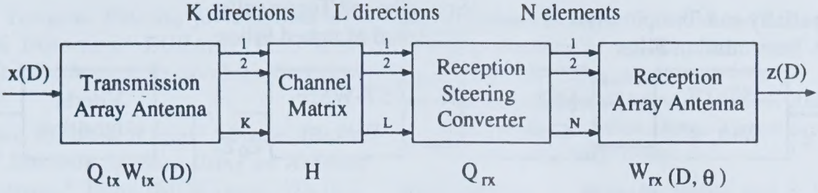


Fig. 8. Channel Including Transmitter and Receiver

component due to ISI. That is

$$W_S(\theta) = \Phi. \tag{36}$$

Using the above ST-WMF, we propose an optimal receiver in the spatial and temporal domains that uses a ST-WMF and a VD which performs MLSE on the TDL array antenna output. As a special case, a receiver with a S-WMF and VD and a receiver with a T-WMF and VD are included.

C Joint Transmitter-Receiver System

Figure 9 shows a MLSE, which is connected to a ST-WMF built using a TDL array antenna. The detection algorithm for received sequence is described as follows.

1. Each antenna element receives signals.
2. Received signals in each antenna element are filtered by a ST-WMF, which is matched to the transmission channel impulse response.
3. The maximum likelihood sequence is estimated for the ST-WMF output.

If AWGN is filtered by the WMF, the noise component at the WMF output becomes correlated. Therefore, the effect of the correlated noise must be removed by the noise whitening filter. Since the noise becomes correlated due to filtering by the WMF, the noise whitening filter is the inverse filter of the ST-WMF. Thus, the difference between the output sequence $z(D)$ and it's estimate $\hat{z}(D)$ is filtered by the noise whitening filter.

D Upper Bound of Symbol Error Rate

The output sequence $z(D)$ of a ST-WMF is represented by the sum of the signal component $y(D)$ and the whitened noise component $n(D)$ as

$$z(D) = y(D) + n(D), \tag{37}$$

where $y(D)$ is described with the l^{th} signal sequence vector $\mathbf{s}_l(D, \theta_l)$ and the array impulse response vector $\mathbf{w}_r(D, \theta)$ as follows

$$\begin{aligned} y(D) &= \mathbf{w}_r^T(D,\theta) \sum_{l=0}^{L-1} \mathbf{r}_l(D,\theta_l) \\ &= f(D)x(D). \end{aligned} \tag{38}$$

The impulse response $f(D)$ of the joint ST-WMF channel which includes a transmitter, a channel and a receiver, is given by

$$\begin{aligned} f(D) &= \mathbf{w}_r^T(D) \sum_{l=0}^{L-1} \mathbf{q}(\theta_l) g_l(D) \\ &= f_0 + f_1 D + \cdots + f_{2\lambda-1} D^{2\lambda-1}, \end{aligned} \tag{39}$$

2λ–1 is the total memory of the joint channel including the transmitter and receiver. The squared minimum Euclidean distance d_{min}^2 for the joint ST-WMF output is defined by

$$d_{min}^2 = ||f||^2. \tag{40}$$

The energy of the impulse response $f(D)$ is defined by

$$||f||^2 \equiv \sum_{i=0}^{2\lambda-1} f_i^2. \tag{41}$$

The symbol error probability $P(e)$ of the proposed optimal receiver in the spatial and temporal domains is bounded by

$$P(e) \le \alpha Q(d_{min}/2\sigma), \tag{42}$$

where α is a small constant and $Q(\cdot)$ is the error function.

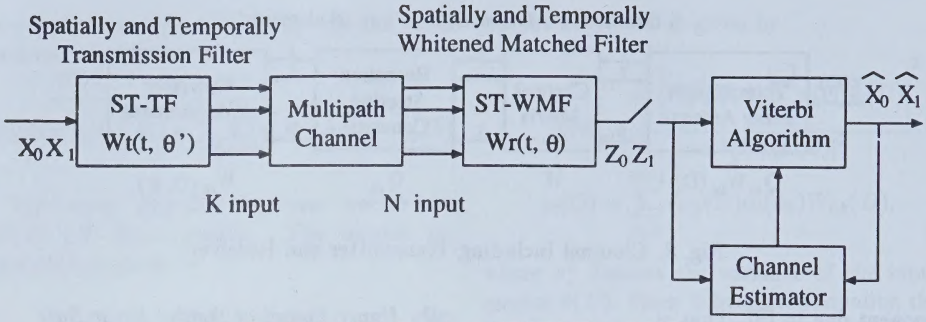

Spatially and Temporally
Transmission Filter
Spatially and Temporally
Whitened Matched Filter
ST-TF
Wt(t, θ')
Multipath
Channel
ST-WMF
Wr(t, θ)
Viterbi
Algorithm
$\widehat{X_0}\ \widehat{X_1}$
$X_0 X_1$
$Z_0 Z_1$
K input
N input
Channel
Estimator

Fig. 9. MLSE Based on ST-WMF and VD

E Acheivable Transmission Rate $\mathcal{R}_{st}$

It is assumed that the receiver would ideally filter out all received signals and noise outside the band $(-W, W)$, and the noise samples are independent and have variance σ^2. The transmission rate $\mathcal{R}$ [bps] is given by

$$\mathcal{R}_{st} = W \log \frac{\sigma_s^2 \sum_{i=0}^{2\lambda-1} |f_i|^2 + \sigma^2}{\sigma^2}, \quad (43)$$

where the input has variance σ_s^2. This formular takes into account the ISI.

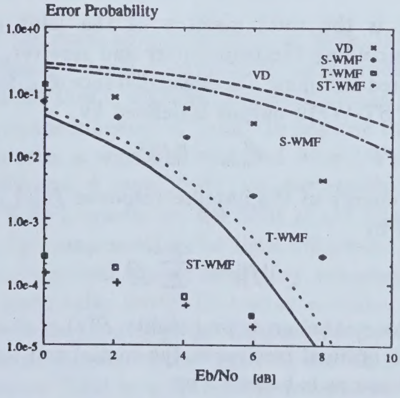


Fig. 10. BER Performance

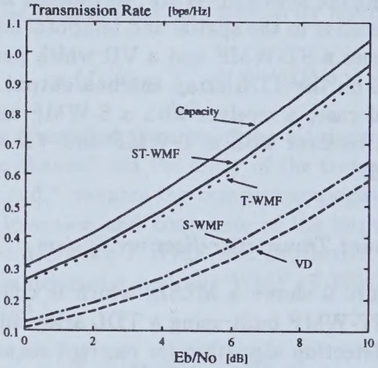


Fig. 11. Acheivable Transmission Rate

VI. CONCLUDING REMARKS

Feasibility of implementing an adaptive array antenna is increasing in higher frequency bands such as millimeter wave band. The spatial and temporal communication theory will be more important to achieve high-speed and highly reliable radio communications. Analysis and optimization of the joint equalization should be acheived for more practical mobile radio channels.

REFERENCES

[1] Shahid U.H.Qureshi : "*Adaptive Equalization,*" Proceedings of the IEEE, Vol. 73, No. 9 Sep. 1985

[2] R. Kohno, H. Imai, S. Pasupathy : "*Adaptive Array Antenna Based on Combination of Spa-*

tial and Temporal Filtering for Channels with Multipath Distortion," EURASIP Fifth European Signal Conference, Proceedings, Barcelona Sep. 1990

3] N.Kuroiwa, R.Kohno, H.Imai : "*Design of a Directional Diversity Receiver Using an Adaptive Array Antenna,*" Trans. IEICE Japan, J73-B-II, No. 11, pp. 755–763 Nov. 1990

4] R.Kohno, H.Wang, H.Imai : "*Adaptive Array Antenna Combined with Tapped Delay Line Using Processing Gain for Spread-Spectrum CDMA Systems,*" PIMRC'92, Boston, Massachusetts 1992

5] M.Nagatsuka, N.Ishii, R.Kohno and H.Imai, "*Array Antenna Based on Spatial Spectrum Estimation Using Maximum Entropy Method,*" IEICE Trans. Commun., vol.E77-B, No.5, pp.624 – 633, May 1994

6] N.Ishii and R.Kohno, "*Spatial and Temporal Equalization Based on an Adaptive Tapped Delay Line Array Antenna,*" IEICE Trans. Commun., vol.E78-B, No.8, pp.1162 – 1169, August 1995

7] R.Kohno, "*Spatial and Temporal Filtering for Co-Channel Interferece in CDMA,*" IEEE International Symposium on Spread Spectrum Tech. & Appli., pp.51 – 60, July 1994

8] R. T. Compton. Jr., "*Adaptive Antennas: Concepts and Performance,*" Prentice Hall, Englewood Cliffs, NJ 1988

9] G. D. Forney, Jr., "*Maximum-likelihood Sequence Estimation of Digital Sequences in The Presence of Intersymbol Interference,*" IEEE Trans. Inform. Theory, vol.IT-18, pp.363–378, May 1972

0] F. R. Magee, jr., J. G. Proakis, "*Adaptive Maximum-Likelihood Sequence Estimation for Digital Signaling in the Presence of Intersymbol Interference,*" IEEE Trans. Inform. Theory, vol.IT-19, pp. 120 – 124, Jan. 1973

1] R. Kohno , H. Imai and M. Hatori , "*Cancellation techniques of co-channel intererence in asynchronous spread spectrum multiple access systems,*" Trans. IEICE, vol. J66-A, no. 5, pp. 416 – 423, May 1983

2] S. Verdu, "*Minimum probability of error for asynchronous Gaussian multiple access channels,*" IEEE Trans. Inform. Theory, vol. IT-32, pp. 85 – 96, Jan. 1986

[13] R. Kohno, H. Imai, M. Hatori and S. Pasupathy, "*Combination of an Adaptive Array Antenna and a Canceller of Interference for Direct-Sequence Spread-Spectrum Mutliple-Acess System,*" IEEE J. Select. Areas Commun., vol. 8, p. 675 – 682, May 1990

[14] R. Kohno, H. Imai, M. Hatori and S. Pasupathy, "*An adaptive canceller of co-channel interference for spread spectrum multiple access communication networks in a power line,*" IEEE J. Select. Areas Commun., vol. 8, p. 691 – 699, May 1990

[15] M. Nagatsuka, R. Kohno and H. Imai, "*Optimal Receiver in Spatial and Temporal Domains Using Array Antenna,*" International Symposium on Information Theory and Its Applications 1994, pp. 893 – 898, Nov. 1994

[16] Shmuel Y. Miller and Stuart C. Schwartz, "*Integrated Spatial-Temporal Detectors for Asynchronous Gaussian Multiple-Access Channels,*" IEEE Trans. on Commun., vol. 43, pp. 396 – 411, Feb./March/April 1995

[17] Rafal Krenz and Krzysztof Wesolowski, "*Comparison of Several Space Diversity Techniques for MLSE Receivers in Mobile Communications,*" IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC'94), vol.II, pp.740–744, Sept. 1994

[18] A. Saifddin, R. Kohno and H. Imai, "*Integrated Co-Channel Interference Cancellation and Decoding Scheme over Fading Multipath Channel for CDMA,*" Trans. IEICE, Vol. J77-B-II, No. 11, pp. 608 – 617, Nov. 1994

[19] A. Saifddin, R. Kohno and H. Imai, "*Cascaded Combination of Cancelling Co-Channel Interference and Decoding of Error-Correcting Codes for CDMA,*" Proc. of IEEE ISSSTA'94, pp. 373 – 377, Oulu, Finland

[20] A. Saifddin and R. Kohno, "*Performance Evaluation of Near-Far Resistant Receiver for DS/CDMA Cellular System over Fading Multipath Channel,*" Trans. IEICE, Vol. E78-B, No. 8, pp. 1136 – 1144, Aug. 1995

[21] A. Saifddin, R. Kohno and H. Imai, "*Integrated Receiver Structure of Staged Decoder and CCI Canceller for CDMA with Multilevel Coded Modulation,*" Europ. Trans. on Telecomm. and Related Technol., Vol. 6, No. 1, pp. 9 – 19, Jan.-Feb., 1995

The Multi-access Fading Channel: Shannon and Delay Limited Capacities

Stephen V. Hanly and David N. Tse
AT&T Bell Laboratories, Murray Hill
{hanly,tse}@research.att.com

Abstract

We use statistical multiplexing and power control to define a notion of delay limited capacity, valid for traffic such as voice and video that cannot tolerate delay. Our capacity bounds the performance of schemes that are robust to changes in the rate of the fading processes.

1 Introduction

Fading poses severe problems in mobile, wireless communications. An important technique currently used to mitigate fading is error control coding. Shannon theory enables us to calculate the maximum possible rate in bits/sec that can be reliably sent on the channel. It is not clear, however, that such Shannon capacities are always useful measures if the traffic is *delay limited.* With block lengths constrained by the delay requirements of the traffic, it is not clear that the fading will always be sufficiently fast to allow averaging over the block length.

In the present paper, we show that a notion of "delay limited capacity" can be defined. We first consider a white Gaussian noise model in which users suffer fading, but users do not have the facility of power control. Rather than average the fading over time, we use statistical multiplexing to average the fading over *users.* In a similar manner to the random coding exponent approach (see for example [2]), we compute a bound on error probabilities consisting of two terms: one, which decays exponentially in time, much like the error exponent of a white noise, non-fading channel, and one that decays exponentially in the number of users. Moreover, the bound depends only on the stationary distribution of the fading process, and not on the rate of the fading process. We use this bound to define the notion of delay limited capacity. To achieve rates above the delay limited capacity, block lengths long enough to average out the (possibly slow) fading are then required. In Section 3, we extend this notion to systems with power control, generalizing the single-user work of Goldsmith [4], in which Shannon capacity is calculated assuming feedback from the receiver to the transmitter about the state of the channel. First we consider decentralized power control, in which users allocate power based on their own level of fading. Finally, in Section 3.3 we show that with optimal centralized power control, in which powers are chosen by the base station as a function of the joint vector of all fading levels, we can define delay limited capacity without the need for statistical multiplexing. We calculate both delay limited capacities and Shannon capacities, and characterize the optimal power control for each case.

In the context of a single user channel, prior work by Ozarow *et. al* [9] has considered the issue of delay limited traffic. Cheng [1] considers the issue of fading in the multi-user context. Both papers take an outage approach.

An interesting conclusion we reach is that to achieve Shannon capacity in a multi-user channel with fading, and with a long-term average power constraint on each user, users *cannot* be transmitting in every time slot. The delay for each user *must* be a function of the rate of fading in the channel. However, it is clear that a delay limited user must be allocated time slots at a rate independent of the rate of fading. We show that bit rates below the delay limited capacity can be achieved by spread spectrum schemes in which each user is allocated every time and frequency slot. In fact, there must be enough "sharing of spectrum" to allow the total instantaneous mutual information in the channel to be split amongst the users according to their rate requirements.

Knopp and Humblet [7] have recently obtained the multi-user Shannon capacity under optimal centralized power control. We will review this in Section 3.3.1. We become aware of this work while preparing the present paper, and the results in the present paper were obtained independently from those in [7].

2 Statistical Multiplexing of Fading

Consider the continuous-time multiple-access channel:

$$Y(t) = \sum_{i=1}^{M} \sqrt{H_i(t)} X_i(t) + Z(t) \tag{1}$$

where M is the number of users, $X_i(t)$ and $H_i(t)$ are the transmitted waveform and the fading process of the ith user respectively, and $Z(t)$ is white Gaussian noise with power spectral density $\eta/2$. Note that in this model, we consider fading effects which are frequency non-selective. We assume that the fading processes for all users are stationary, ergodic, statistically identical and independent of each other. We also assume that there are finite number of fading levels $\gamma_1 < \gamma_2 < \ldots < \gamma_K$, such that $\mathbb{P}(H_i(0) = \gamma_k) = \phi_k$. In practice, the fading process takes on a continuum of values and this can be taken as a discrete approximation to simplify the mathematics.

A total bandwidth of $2MW_0$ is available for communication (i.e. $2W_0$ per user.) Since we are dealing exclusively with frequency non-selective effects, it suffices to assume that the transmitted waveforms are bandlimited to $[-MW_0, MW_0]$. Each user is also subjected to an average transmitted power constraint of $\bar{P}$. While the decoder is assumed to have complete knowledge of the realizations of the fading processes, the transmitters know only their statistics. (In the next section, we will consider the case when the transmitters also have access to the realizations of the fading processes and can perform power control based on the instantaneous state of the channel.)

The capacity region for this multiaccess channel is well-known; it is given by:

$$\mathcal{C} = \{(R_1, \ldots, R_M) : \sum_{i \in \mathcal{L}} R_i \leq \mathbb{E}_{\vec{H}}[MW_0 \log(1 + \frac{\bar{P}}{M\eta W_0} \sum_{i \in \mathcal{L}} H_i)], \forall \mathcal{L} \subset \{1, \ldots, M\}\}$$

where $\vec{H} = (H_1, \ldots, H_M)$ and H_i's are i.i.d. random variables having the stationary distribution of the fading processes. We are interested in the case when each user wants to transmit at equal rate; by symmetry, the equal-rate capacity (per user) is given by:

$$C = \mathbb{E}_{\vec{H}}[W_0 \log(1 + \frac{\bar{P}}{M\eta W_0}\sum_{i=1}^{M} H_i)] \tag{2}$$

For the purpose of focusing on the essential ideas, we shall first consider a simple class of fading processes, having only two fading levels γ_1 and γ_2 ($\gamma_1 < \gamma_2$), and that $\mathbb{P}(H_i = \gamma_1) = \phi$, $\mathbb{P}(H_i = \gamma_2) = 1 - \phi$, The equal-rate capacity for this fading model is

$$C(M) = \mathbb{E}_{M_1}[W_0 \log(1 + \frac{\bar{P}(\gamma_1 M_1 + \gamma_2(M - M_1))}{M\eta W_0})] \tag{3}$$

where M_1 is a random variable with Binomial distribution $\mathrm{Bin}(M, \phi)$; it is the steady-state number of users who are fading at level γ_1. An intuitive understanding of this result can be obtained by viewing the capacity in terms of a time average of mutual information ([3]), the rate of flow of which can be viewed as a random process depending on the fading levels of the users. Specifically, if $M_1(t)$ is the number of sources fading at level γ_1 at time t, then the instantaneous rate of flow of joint mutual information between the sources and the receiver can be thought of as:

$$MW_0 \log(1 + \frac{\bar{P}(\gamma_1 M_1(t) + \gamma_2(M - M_1(t)))}{M\eta W_0}) \tag{4}$$

(This assumes that the transmitted waveforms are independent white Gaussian processes with power $\bar{P}$.) Thus the amount of mutual information per source averaged over a time interval $[0, T]$ is

$$\frac{1}{T}\int_0^T W_0 \log(1 + \frac{\bar{P}(\gamma_1 M_1(t) + \gamma_2(M - M_1(t)))}{M\eta W_0})dt$$

As $T \to \infty$, this quantity converges to (3), since the process $\{M_1(t)\}$ is stationary and ergodic. From this interpretation, one can see that to transmit information reliably at a rate close to the capacity (3), one must code over a duration T significantly larger than the time-scale of the fluctuations of the fading processes $\{H_i(t)\}$. This is so that one can "accumulate" sufficient mutual information per unit time, close to the ergodic average, before decoding.

In many situations, it is unrealistic to code over such a time-scale, especially when there are strict delay requirements (such as for real-time services like voice or video) and some components of the fading are slow.

The capacity (3) is then not a very meaningful measure in those situations. It is then of interest to ask what can be achieved if codeword lengths are not necessarily long enough to average the fading over time. It turns out that if there are sufficiently large number of sources, then in some cases one can still get close to the Shannon capacity (3) without the need to average the fading over time. Instead, the averaging is done over the fading of the different *users* at a given time.

Let us look more carefully at the situation when there are many sources. By law of large numbers, the fraction of sources at γ_1 at time t, $\frac{M_1(t)}{M}$, converges to ϕ as M becomes large. From (3), the equal-rate Shannon capacity C_M converges to

$$C = W_0 \log(1 + \frac{\bar{P}(\gamma_1\phi + \gamma_2(1 - \phi))}{\eta W_0})]$$

¿From (4), the rate of flow of joint mutual information between the sources and the receiver at any time t is given by:

$$MW_0 \log(1 + \frac{\bar{P}(\gamma_1\phi + \gamma_2(1-\phi))}{\eta W_0})] + o(M) = MC + o(M)$$

This suggests the possibility of each user transmitting reliably at rate close to C, by "splitting" the mutual information equally between the sources. Large delays need not incurred since we do not have to average the fades over time to get this amount of mutual information. We will make precise below when this can be done.

The continuous-time baseband system (1) is sampled at the Nyquist rate $\frac{1}{2MW_0}$ and the resulting discrete-time system is:

$$Y_n = \sum_{i=1}^{M} \sqrt{H_{in}} X_{in} + Z_n$$

where $y_n = y(\frac{n}{2MW_0})$, etc. Suppose each source i codes over a block length of $2MW_0T$ symbols, where T is the delay in seconds, using a codebook $\mathcal{C}_i$ of size 2^{RT} (i.e. at rate R bits per second). The following result hold.

Theorem 1 *For any rate $R < C_d$, where*

$$C_d \equiv \min\{\frac{W_0}{\phi}\log(1 + \frac{\bar{P}\gamma_1\phi}{\eta W_0}), W_0\log(1 + \frac{\bar{P}(\gamma_1\phi + \gamma_2(1-\phi))}{\eta W_0})\}$$

and for any $\epsilon > 0$, one can choose M,T and codebooks $\mathcal{C}_1, \ldots, \mathcal{C}_M$ of size 2^{RT}, **independent of the correlation structure of the fading processes** *$H_i(\cdot)$'s but only dependent on the stationary distribution, such that the probability of decoding some user incorrectly is less than ϵ. Conversely, if a rate R is achievable in the above sense, then necessarily $R \leq C_d$.*

Thus, the coding delay to get a small probability of error is independent of how slow the rate of change of the fading is. We call C_d the *delay limited* capacity, to distinguish it from the Shannon capacity. We note that if

$$\frac{W_0}{\phi}\log(1 + \frac{\bar{P}\gamma_1\phi}{\eta W_0}) \geq W_0\log(1 + \frac{\bar{P}(\gamma_1\phi + \gamma_2(1-\phi))}{\eta W_0}) \tag{5}$$

then $C_d = C_\infty$ and we can transmit at rates close to Shannon capacity without the need of time averaging over fades.

Although we shall not present proofs in this paper, it is instructive to look at the form of the bound we obtained on the error probability:

$$p_e \leq \frac{M\exp(-TE(\delta, R) + 1)}{1 - M\exp(-TE(\delta, R) + 1)} + 2MW_0T\exp(-MI(\delta)).$$

and this bounds holds for all $\delta > 0$. The exponent $E(\delta, R)$ is positive for $R < C_d - \delta$. The first term decays exponentially with T and is a consequence of averaging over the white noise. It is derived from random coding exponents associated with certain Gaussian channels. The second term decays exponentially with M and is the large deviations probability of not getting sufficient mutual information over the time interval $[0, T]$.

We can give another interpretation of the quantity C_d: it can be shown that C_d is also the equal-rate capacity of a multi-access system where a fraction ϕ of the users are *always* at fading level γ_1 and $(1-\phi)$ of the users are *always* at level γ_2 (Theorem 6.4 of [5]). The conclusion is therefore that if the number of sources is large and the delay constraint is stringent, we can essentially view the original system as one in which the fading levels of individual sources never change but the fraction of sources at each level is the same as that in the equilibrium distribution of the original system.

The above result can be generalized to processes of arbitrary number of fading levels, $\gamma_1 < \ldots < \gamma_K$. The delay limited capacity is given by:

$$C_d = \min_{k=1,\ldots,K} \frac{W_0}{\sum_{j=1}^{k} \phi_j} \log\left(1 + \frac{\bar{P} \sum_{j=1}^{k} \phi_j \gamma_j}{\eta W_0}\right) \tag{6}$$

Again, this has the interpretation of the equal-rate capacity of a system where $\phi_k M$ users are always at fading level γ_k.

3 Power control from channel feedback

3.1 Decentralized power control

In the single user channel studied in Goldsmith [4], a user is allowed to adjust its transmitter power based on the current level of fading it is experiencing. In the present section, we extend this to the multi-user context, but still only allow a user to power control based on its own fading level. We consider the centralized approach in which the power control is a joint optimization over all users' fading levels in Section 3.3. The present power control is motivated by the open loop power control for CDMA in the IS95 standard [10]. Each mobile measures the strength of the cell site pilot, of known transmit power, and deduces from this the level of fading.

Let the power control of each transmitter be given by $\mathcal{Q}(\gamma_k) = Q_k, k = 1, \ldots, K$. (By symmetry, we can assume that all transmitters use the same control.) We impose the long term power constraint

$$\sum_{k=1}^{K} \phi_k Q_k = \bar{P} \tag{7}$$

We can specify any power control by the associated vector $(Q_1, Q_2, \ldots, Q_k)$.

Given any power control $\mathcal{Q}$, there is a corresponding Shannon capacity.

$$C_M(\mathcal{Q}) = \mathbb{E}_{\mathcal{M}}\left[W_0 \log_2\left(1 + \frac{\sum_{k=1}^{K} M_k \gamma_k Q_k}{\eta W}\right)\right] \tag{8}$$

where $\mathcal{M} = (M_1, \ldots, M_K)$ is a Multinomial random vector with parameters $(M, \phi_1, \ldots, \phi_M)$.

The Shannon capacity of the channel subject to the power constraint (7) is obtained by maximizing (8) over all $\mathcal{Q}$ satisfying (7). The optimal power control in the limit as $M \to \infty$ can be computed explicitly and is given by:

$$Q_k^\star = \begin{cases} \bar{P}/\phi_K & k = K \\ 0 & k \neq K \end{cases} \tag{9}$$

The corresponding limiting capacity is given by

$$C \equiv W_0 \log\left(1 + \frac{\gamma_K \bar{P}}{\eta W_0}\right) \tag{10}$$

The interpretation is that each user waits until its fading conditions are best possible before sending any information.

This Shannon capacity is not delay limited, since a user must wait until its own fading level is best possible. To compute the delay limited capacity for an average power constraint $\bar{P}$, we first obtain the delay limited capacity $C_d(\mathcal{Q})$ for an arbitrary power control policy $\mathcal{Q}$. This is a generalization of the result in (6), and can be shown to be:

$$C_d(\mathcal{Q}) = \min_{\emptyset \neq \mathcal{L} \subseteq \{1,2,\ldots,K\}} \frac{W_0}{\sum_{k\in\mathcal{L}} \phi_k} \log\left(1 + \frac{\sum_{k\in\mathcal{L}} \phi_k \gamma_k Q_k}{\eta W_0}\right), \tag{11}$$

To compute the optimal power control which minimizes this capacity subject to an average power constraint, we look at the equivalent problem of minimizing the average power to achieve a given delay limited capacity R:

$$\min_{\mathcal{Q}: C_d(\mathcal{Q}) \geq R} \sum_{k=1}^{K} \phi_k Q_k \tag{12}$$

To solve this optimization problem, we need the following lemma.

Lemma 1 *Let $c_1 \geq c_2 \geq \ldots \geq c_n$ and $R_1, R_2, \ldots, R_n$ be given positive numbers. Consider the problem:*

$$\min \sum_{j=1}^{n} c_j x_j \quad s.t. \sum_{j\in\mathcal{L}} R_j \leq W_0 \log\left(1 + \frac{\sum_{j\in\mathcal{L}} x_j}{\eta W_0}\right) \quad \forall \mathcal{L} \subseteq \{1,2,\ldots,n\} \tag{13}$$

Then the optimizing x^* is specified **recursively** *by the equations*

$$R_i = W_0 \log\left(1 + \frac{x_i^*}{\eta W_0 + \sum_{j=1}^{i-1} x_j^*}\right) \quad i = 1,2,\ldots,n \tag{14}$$

This result has an interesting interpretation. Suppose that there are n users and user i requires bit rate R_i. Suppose in addition that the fading level for user i is γ_i and that this is fixed for all time. Since it is fixed, it is probably better to call it the path gain from user i to the receiver. Then under the correspondence $c_i = \frac{1}{\gamma_i}$ $i = 1,2,\ldots,n$, the above problem is precisely the problem of minimizing the sum of the transmit powers subject to the bit rate constraints. Moreover, Lemma (14) shows that the optimal solution is to do successive decoding in descending order of the path gains, i.e. user i is decoded treating users $1, 2, \ldots, i-1$ as noise, its signal then subtracted from the received waveform before decoding user $i-1$. An interesting observation is that the optimal decoding order is *independent* of the bit rate requirements of the users, depending only on the path gains.

Applying the lemma, we can solve the optimization problem (12). The required delay limited capacity can then be solved for by setting R in (12) such that the average power equals the $\bar{P}$. Thus, the delay limited capacity is a solution to a system of equations which can be solved iteratively.

Theorem 2 *For an average power constraint $\bar{P}$, the delay limited capacity C_d and the associated optimal power control $\mathcal{Q}^*$ are unique solutions to the equations:*

$$\begin{cases} C_d = W_0 \log\left(1 + \dfrac{\phi_k \gamma_k Q_k^*}{\eta W_0 + \sum_{j=1}^{k-1} \phi_j \gamma_j Q_j^*}\right) \quad k = 1,2,\ldots,K \\ \sum_{k=1}^{K} \phi_k Q_k^* = \bar{P} \end{cases} \tag{15}$$

3.2 Example: two fading levels

To gain more insight into these results, we consider the simple case of two fading levels: γ_1 and γ_2. The delay limited capacity, C_d, occurs precisely when

$$W_0 \log_2 \left(1 + \frac{\phi\gamma_1 Q_1(Q_2) + (1-\phi)\gamma_2 Q_2}{\eta W_0}\right) = \frac{W_0}{\phi} \log \left(1 + \frac{\phi\gamma_1 Q_1(Q_2)}{\eta W_0}\right)$$

by Theorem 2. We denote the optimal power control to achieve C_d by $Q^\star$.

To provide insight into this result, let us consider a different model, without fading, but with the same power control as above. There are M users, and a deterministic proportion ϕ of these users are *always* in state γ_1, with the rest *always* in state γ_2. All users in the first group achieve a common rate denoted by R_1, and those with the higher gain achieve R_2. We consider three different cases of Q_2, defined by the levels $Q_2^{(a)}, Q_2^{(b)}, Q_2^{(c)}$. We choose these values such that $0 < Q_2^{(a)} < Q_2^\star, Q_2^{(b)} = Q_2^\star, Q_2^\star < Q_2^{(c)} < \frac{\bar{P}}{1-\phi}$ and depict the feasible rate regions $\mathcal{R}^{(a)}, \mathcal{R}^{(b)}, \mathcal{R}^{(c)}$ in Figure 1. Note that in the limit as $M \uparrow \infty$, $\frac{\bar{P}}{1-\phi}$ is the optimal power control to achieve the limiting Shannon capacity C of the channel. This follows from (9).

Let us focus first on equal rate points on the equal rate line depicted in the figure. For equal rates, capacity increases from (a) to (b), but decreases from (b) to (c), because the constraint corresponding to just the users in state γ_1 has started to bite. Case (b) represents the maximum equal rate capacity, which coincides with C_d. By increasing Q_2 from $Q_2^{(b)}$ to $Q_2^{(c)}$, users in state γ_1 suffer a decrease in rate, but a rate pair $(R_1^{(c)}, R_2^{(c)})$ can be achieved for which $R^{(c)} \equiv \phi R_1^{(c)} + (1-\phi)R_2^{(c)} > C_d$ With power allocation $Q^{(c)}$ in our *fading* model, users switch between γ_1 and γ_2. All users can achieve the common rate $R^{(c)}$, which we have noted to be greater than C_d, by time-sharing between $R_1^{(c)}$ and $R_2^{(c)}$. An important point is that if the fading is slow, then the delay can be considerable. The limiting Shannon capacity C is also depicted in the figure.

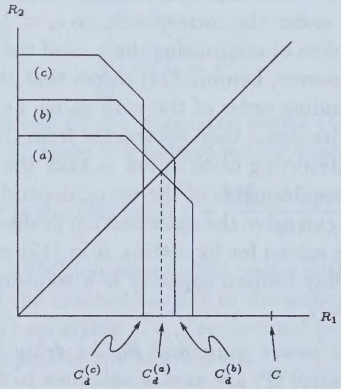


Figure 1: Rate regions as a function of Q_2

3.3 Centralized power control

In this section, we allow joint power control, in which each user's power level $P_i(t)$ at time t is a function of *all* the fading levels $H_1(t), \ldots, H_M(t)$ in the system, i.e. $P_i(t) = \mathcal{P}_i(H_1(t), \ldots, H_M(t))$ for a control policy $\mathcal{P}$.

3.3.1 Shannon capacity

Using (2), the Shannon capacity of the channel under optimal centralized power control, subject to the long-term average power constraint $\bar{P}$, can be shown to be:

$$\begin{aligned} C_M &= \max_{\mathcal{P}} \mathbb{E}_{\vec{H}} \left[\log \left(1 + \frac{\sum_{i=1}^{M} \mathcal{P}_i(\vec{H}) H_i}{M \eta W_0} \right) \right] && (16) \\ \text{s.t.} \qquad & \mathbb{E}_{\vec{H}}[\mathcal{P}_i(\vec{H})] = \bar{P} \quad i = 1, 2, \ldots, M && (17) \end{aligned}$$

where the maximization is over all power control policies $\mathcal{P}$.

The optimal power control for this problem is given by

$$\mathcal{P}_i^*(h_1, \ldots, h_M) = \begin{cases} \frac{1}{\lambda} - \frac{1}{h_i} & \text{if } h_i > \lambda, h_i = \max_j h_j \\ 0 & \text{else} \end{cases}$$

and λ is chosen such that the average power constraint is satisfied. Thus, at any one time, only the user with the best channel transmits and the power he uses is determined by a water-filling solution. This result is due to Knopp and Humblet [7].

As M grows large, it can be shown that the limiting capacity under this optimal control is given by:

$$\lim_{M \to \infty} C_M = W_0 \log(1 + \frac{\bar{P} \gamma_K}{\eta W_0})$$

This is due to the fact that when there are many sources, with high probability there will be a user at the best fading level γ_K at any time. Comparing this with eqn. (10), we see that this is also the limiting capacity under the optimal *decentralized* power control. Thus for large M, decentralized power control performs as well as centralized control in achieving Shannon capacity.

3.3.2 Delay limited capacity

The Shannon capacity obtained above is not delay limited, since each user has to wait until it has the best fading level among all users before it can send any information. We now turn to the characterization of the delay limited capacity under centralized power control.

Recall that in the cases when there is no power control or when the power control is decentralized, our notion of delay limited capacity is defined only in the asymptotic regime of large number of users. This is because statistical multiplexing is needed to assure that there is sufficient mutual information most of the time. In contrast, with centralized control, we can coordinate the powers of all users so that there is sufficient mutual information *at all times*. Thus, we can define the delay limited capacity $C_d(M)$ for a fixed number of users M: it is the best equal rate that one can achieve in the sense of getting arbitrarily small decoding error probability with delays that are independent of the rate of change of the fading process.

Theorem 3 *The delay limited capacity* $C_d(M)$ *is the largest* R *such that for some power control* $\mathcal{P}$,

1. $\forall\, \vec{h} = (h_1, \ldots, h_M) \quad |\mathcal{L}|R \leq MW_0 \log\left(1 + \frac{\sum_{i\in\mathcal{L}} \mathcal{P}_i(\vec{h})h_i}{M\eta W_0}\right), \quad \mathcal{L} \subseteq \{1, 2, \ldots, M\}$

2. $\mathbb{E}_{\vec{H}}[\mathcal{P}_i(\vec{H})] = \bar{P}, \quad i = 1, 2, \ldots, M.$

It can be shown that for any joint fading state $\vec{h}$, the optimal power allocation $\mathcal{P}^\star(\vec{h})$ is the solution of the following problem:

$$\min \sum_{i=1}^{M} P_i \;\; s.t. |\mathcal{L}|C_d(M) \leq MW_0 \log\left(1 + \frac{\sum_{i\in\mathcal{L}} P_i h_i}{M\eta W_0}\right), \; \mathcal{L} \subseteq \{1, \ldots, M\}$$

Using Lemma (1) again, we obtain the following characterization of $C_d(M)$ and the optimal control.

Theorem 4 1. *For all joint fading states* $\vec{h}$, *the delay limited capacity* $C_d(M)$ *and the optimal power allocation* $\mathcal{P}^\star(h)$ *satisfy:*

$$C_d(M) = MW_0 \log\left(1 + \frac{h_{u_i}\mathcal{P}^\star_{u_i}}{M\eta W_0 + \sum_{j=1}^{i-1} h_{u_j}\mathcal{P}^\star_{u_j}}\right), \quad i = 1, 2, \ldots, M$$

where $h_{u_1} \leq h_{u_2} \leq \ldots h_{u_M}$.

2. $C_d(M)$ *is uniquely characterized by the equation*

$$\mathbb{E}_{\vec{H}}[\mathcal{P}^\star_1(\vec{H})] = \bar{P}$$

Note that if the fading is sufficiently slow that it does not change during the codeword length, then $C_d(M)$ can be achieved by successive decoding, starting from the user at the best fading level.

If users take on the same fading level, the optimal allocation of powers is not unique. Denote the set of users at level γ_k by $\mathcal{M}_k(\gamma)$, $k = 1, 2, \ldots, K$. Given a joint fading state $\vec{h}$, define

$$\mathcal{Q}^\star_k(\vec{h}) \equiv \frac{1}{|\mathcal{M}_k|} \sum_{i\in\mathcal{M}_k} \mathcal{P}^\star_i(\vec{h}).$$

We can assign an averaged power allocation $\hat{\mathcal{P}}_i(\vec{h}) = \mathcal{Q}^\star_k(\vec{h})$ if user i is at fading level k. It can be shown that this power allocation is also optimal. Note that under this control, the power allocated to a user depends only on its own fading level and the empirical distribution of users in the different fading levels at that time. As the number of users gets large, the empirical distribution converges to the stationary distribution of the fading process, so it can be expected that decentralized power control is asymptotically optimal. Indeed, this is so.

Theorem 5 *As the number of users* M *grows,* $C_d(M)$ *converges to the delay limited capacity under optimal decentralized power control, as characterized by Theorem 2.*

Thus, when there are large number of users, it is nearly optimal for each user to allocate power as a function of its own fading level only. This is of particular interest, since the rate of power updates under optimal centralized power control increases linearly with the number of users, and this is an undesirable feature.

4 Conclusions

We have defined a notion of delay limited capacity for traffic with stringent delay requirements. This can be accomplished by a centralized power control to completely mitigate the fading. We also define a delay limited capacity when there is no power control, or when there is a decentralized power control. In this case, we use statistical multiplexing; a large number of users are required.

A crucial feature of all delay limited capacities is that they are achieved by spread spectrum schemes. Indeed, we have shown that the delay limited capacity with power control is achieved by an "successive decoding type" scheme. The main reason is that spread spectrum allows the instantaneous total mutual information in the channel to be split equitably amongst the users. Spread spectrum is also necessary to obtain statistical multiplexing.

We have also characterized the Shannon capacities of the channels and the power control schemes that achieve these capacities. These capacities may be relevant for certain types of data traffic such as emails or faxes in which delay is not a concern. A feature of the optimizing schemes here is that users are not permitted to send data all the time, but only at certain instants when the channel is good.

References

[1] Cheng R. (1993) On capacity and signal waveform design for slowly fading CDMA channels. *Proc. of 31st Allerton Conf.*, Monticello, Illinois, pp. 11-20.

[2] Gallager R. (1968) *Information theory and reliable communication*, Wiley.

[3] Gallager R. (1994) An inequality on the capacity region of multiaccess fading channels. in *Communications and Cryptography - Two sides of one tapestry*, Kluwer, pp. 129-139.

[4] Goldsmith A. (1994) Design and performance of high-speed communication systems over time-varying radio channels *Ph. D. thesis, University of California, Berkeley*, September.

[5] Hanly, S.V. (1993) Information capacity of radio networks *Ph. D. Thesis, Cambridge University* August.

[6] Hanly S.V., Whiting P.A. (1994) Constraints on capacity in a multi-user channel *International symposium on information theory* Trondheim, Norway.

[7] Knopp R., Humblet P.A. (1995) Information capacity and power control in single-cell multiuser communications, *Int. Conf. on Communications*, Seattle, Wash., June.

[8] Rimoldi B., Urbanke R. (1994) Onion peeling and the Gaussian multiple access channel *submitted to IEEE Trans. Inform. Theory*

[9] Ozarow L. H., Shamai S., Wyner A. D. (1994) Information-theoretic considerations for cellular mobile radio.

[10] IS95 standard, Mobile station-base station compatibility standard for dual-mode wideband spread spectrum cellular systems.

Evaluation of Field Measurements From An Experimental Slow Frequency Hop Personal Communication System

Dominic M. Tolli, Phillip D. Rasky, Greg M. Chiasson, and James F. Kepler

Communication Systems Research Laboratory
Motorola, Inc.
1301 East Algonquin Road
Schaumburg, IL 60196 USA

Abstract

A prototype personal communication system utilizing both slow frequency hopping and time-division multiple access is investigated. The performance of this prototype is verified using both simulation-based and field test results. Field measurements are presented which demonstrate significant gains available through the prototype's use of a combination of space, frequency and time diversity. In particular, considerable gains are observed during field operation in the presence of slow Rayleigh fading.

I. Introduction

A communication system employing the hybrid access technique of slow frequency hopping and time-division multiple access (TDMA) has been investigated for high-mobility personal communications in the 1.8 GHz band. This 500 kbps system was designed with a complementary set of diversity techniques so as to allow robust operation on fading channels characteristic of propagation in cells of up to several miles in radius. Key to the link's performance are frequency diversity resulting from the slow frequency hop access, time diversity resulting from interleaved error-correction coding, and space diversity resulting from a dual antenna/receiver configuration. In simulations, the combination of these three diversity techniques allows the receiver to resist delay spread on the order of 1 μsec and to gracefully degrade with more severe multipath. A detailed discussion of the performance analysis for this proposal is contained in [1]. This reference provides the motivation for the hybrid access technique to support a robust, economical, high capacity radio telephone system capable of supporting data services at rates up to 144 kbps.

In order to verify the concepts that drove the system design with actual field measurements, a prototype slow frequency hop (SFH) TDMA system was constructed. Utilizing this prototype, a field testing effort has been conducted with two primary objectives: the accurate characterization of the PCS propagation environment and the assessment and validation of the technologies incorporated into the system. To validate the specific physical layer design, objective link level statistics are compiled versus received signal strength and related to real-time estimates of RMS delay spread and first order statistics on the envelope of the received signal. The performance benefits attainable from the use of frequency, space and time diversity are investigated for both the downlink (outbound) and uplink (inbound).

II. Prototype System Overview

The SFH PCS prototype is designed around a 10 slot TDMA structure. A TDMA frame is 2 msec in duration. A single traffic channel having a throughput of 16 kbps is mapped onto a single slot of the TDMA carrier. The system frequency hops coincident with the TDMA frame structure resulting in a 500 Hz hop rate. Fig. 1 shows the slot structure and the corresponding mapping of the transmit and receive slots onto the TDMA frame structure from the perspective of the mobile station when supporting a 32 kbps service using two time slots. The information stream is forward error correction (FEC) encoded with a rate 1/2 convolutional code and interleaved over a span of 40 msec. The modulation technique is QPSK with raised-cosine filtering. The discrete channelization of the hopping carrier is 400 kHz. A hopping bandwidth of 10 MHz is used for the field results presented in this paper. A summary of the physical layer attributes of the SFH prototype is shown in Table I.

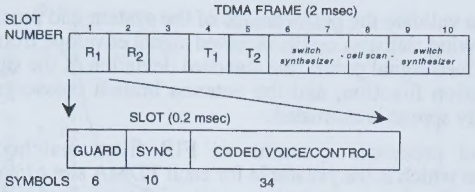


Fig. 1. TDMA frame and slot structure for the hybrid SFH/TDMA PCS.

As mentioned in the introduction, the SFH PCS prototype was designed to be quite robust against the impairments of the harsh mobile radio channel. To support this objective, a two-port receiver structure is used to provide antenna diversity. The QPSK symbols are recovered and combined at baseband with burst-coherent demodulation based on the complex channel gain measured by correlating to the synchronization preamble at the beginning of each slot. Since the transmission is subject to rapidly changing delay spread, the sampling instant of the symbols in each time slot is independent between receiver branches and from slot-to-slot. This effectively provides a selection diversity mechanism over multipath. A more detailed description of the design and function of the prototype can be found in [2].

Field Test Considerations

The full-duplex transceiver hardware is suitably constructed for vehicular field testing. The base and mobile units can transmit a peak power of 800 mW and utilize dual omnidirectional antennas. The mobile station is deployed in a test van with roof-mounted antennas. For the field operations described in this paper, the base antenna site has been deployed at a height of approximately 90 feet at the Motorola Center in Schaumburg, Illinois (suburban Chicago). The terrain in this area can be classified as a dense suburban environment. The area surrounding the base site over a radius of several miles includes residential neighborhoods (not more than 2 stories high), business and industrial parks, shopping malls, and office buildings. There are approximately 15-20 buildings of 10 to 20 stories in height to the south and east of the base site and at a distance greater than one mile away. For both the base and mobile unit, the antenna pairs are separated by approximately 5 feet.

Considerable special purpose testing software has been integrated into the SFH prototype units to allow both subjective and objective quality assessments of the full-duplex link. The slow frequency-hop prototype is capable of logging statistics on the signal received by each of the diversity branches and bit and frame error rate statistics [3]. In this way, the transceiver can be

Table I SFH/TDMA PCS System Specifications.

Frequency Band:	Inbound: 1899-1929 MHz Outbound: 1949-1979 MHz
Channel Spacing	400 kHz
Channel Bit Rate	500 kbps
Frequency Hopping Rate	500 hops/s
Frame Duration	2.0 msec
Time Slots/Frame	10
Duplex Method	FDD
Modulation	QPSK
Pulse Shaping	Raised cosine, α=0.5
Receiver Gain Control	Hard-limiting
Antenna Diversity	Pseudo max-ratio
Demodulation	Burst coherent
Transmit Power	800 mW
Channel Coding:	
Type	Rate 1/2, K=6 convolutional
Interleaving Span	40 msec

simultaneously used to validate the performance of the system and provide channel sounding capabilities. The following statistics on the received signal envelope from each antenna branch are measured: the received signal power, the standard deviation of the signal power, the fading envelope auto-correlation function, and the antenna branch crosscorrelation function. In addition, the RMS delay spread is estimated.

The digital baseband processor contains an FIR filter matched to the 10-symbol synchronization pattern which is the preamble for each TDMA slot. In addition to its function of sounding the amplitude, phase, and timing epoch of the received signal, the matched filter output facilitates the estimation of the RMS delay spread. This is calculated from a power-delay profile accumulated over a measurement interval. The measured RMS delay spread is subject to bias from noise and, due to the symbol rate and filtering, the breadth of the autocorrelation function of the preamble. In order to remove this bias, an estimator which is a function of the measured delay spread and signal-to-noise ratio was developed to more accurately reflect the true RMS delay spread. The estimator was trained in the laboratory utilizing a multipath channel simulator. The resulting estimate of RMS delay spread is accurate to within 0.5 μsec for channels having an RMS delay spread value between 0 and 6 μsec.

During the field testing, all the aforementioned statistics are logged to a file on a notebook computer together with information from a Global Positioning System (GPS) receiver integrated into the mobile station apparatus. The GPS information consists of the time and the mobile velocity and location. Measurement intervals at the base and mobile are synchronized so that the uplink and downlink performance can be compared for close inspection.

III. Field Data and Analysis

Channel Characterization

In order to thoroughly test the SFH PCS prototype, a large number of routes were driven over a region within three miles of the base site. Within this coverage area, several test routes were employed to provide representative channel conditions which could be analyzed to characterize the system's performance. One route approximates a circle with a two mile radius around the base site. The propagation over this route exhibits a broad range of fading statistics since the vehicle speed typically ranges from 20 to 55 mph. Multipath propagation conditions with significant amounts of delay spread are also encountered along this route. As a result, this route permitted an assessment of the performance of the prototype system over a wide range of conditions. A further description of this route, along with presentation and analysis of the field data encountered can be found in [6].

A second route was chosen so as to permit controlled, slow speed testing. It encompasses several square blocks in a residential neighborhood within a mile of the base antenna. This route provided the setting for the data presented in this paper. Field data has shown that the RMS delay spread over this route is predominantly less than 1 μsec, so multipath greater than the order of a signal period is not a significant issue. In addition, since the area is almost completely shadowed by dense vegetation, no line-of-sight path is available between the base and the mobile, and the route is expected to provide a Rayleigh-fading propagation environment. The classical Rayleigh-fading spectrum described in [4] provides a model for this propagation condition.

To verify the existence of a propagation environment with Rayleigh-fading properties, statistics on the coefficient of variation of the envelope of the received signal were accrued on the test route. As described in [3], the coefficient of variation, which is the standard deviation of the signal normalized by the mean signal strength, is useful in assessing the degree to which the fading process matches Rayleigh or Rician fading. Fig. 2 plots the coefficient of variation over a range of Rician parameters, K, defined as the ratio of the power in the direct ray to that in the diffuse rays. When the Rician parameter equals zero, the fading is purely Rayleigh, and as indicated in the figure, the coefficient of variation equals 0.52. If a direct, line-of-sight path exists, the fading process becomes Rician, and as the power in the direct ray increases, the value of the coefficient decreases. Fig. 3 depicts the probability density function for the coefficient of variation measured on the residential route. As previously discussed, in this area very few line-of-sight paths to the transmitter exist; therefore, the fading is expected to be

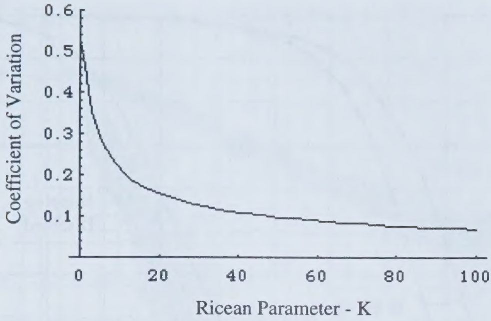


Fig. 2. Coefficient of variation versus the parameter, K.

Rayleigh in nature. Fig. 3 indicates that this hypothesis is reasonable, with the large majority of the coefficients falling in the 0.4 to 0.55 range.

In order to gain further insight into the nature of the propagation environment, statistics are also accrued on the correlation coefficient of the envelope of the received signal. This statistic is of benefit in determining the degree of correlation in the fading process between frames or equivalently, the extent to which frequency hopping has decorrelated the slow fading channel, recalling that there is one frequency hop per frame.

Fig. 4 shows the cumulative distribution function of the correlation coefficient for data gathered while traversing the residential route at low speeds (3 mph). Correlation coefficient curves for a lag of one frame and curves for the coefficient averaged over twenty frames, the span of the interleaver, are depicted. The lag-one coefficient indicates the degree of correlation in the fading process between consecutive frames, while the average coefficient roughly indicates the aggregate level of correlation observed by the forward error correction code, which follows the deinterleaver. Both types curves are depicted for frequency hopped and non-hopped systems.

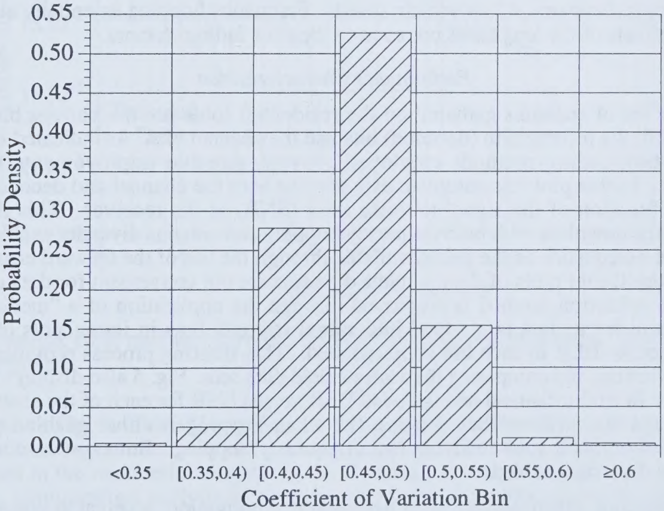


Fig. 3. Probability density function for the coefficient of variation in a predominantly shadowed environment.

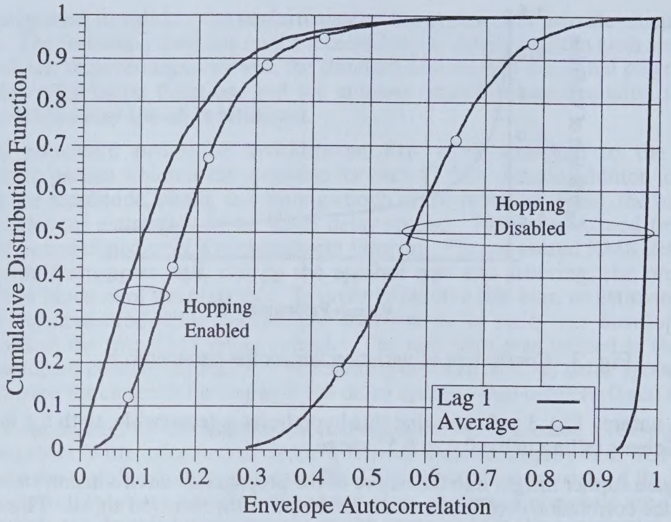


Fig. 4. CDFs of autocorrelation values acquired on the residential route.

As the figure indicates, frequency hopping provides a substantial degree of decorrelation on the slow fading channel. With hopping enabled, over 90% of the correlation coefficients are less than 0.3. In contrast, with hopping disabled, as indicated by the non-hopped, lag-one coefficient, the envelope correlation for consecutive frames is greater than 0.9 over all of the data collected. While the correlation coefficient averaged over the span of the interleaver is somewhat lower, primarily due to the effects of time decorrelation and variations in log-normal shadowing, the channel observed by the error correction code is still substantially more correlated in the non-hopped case. These phenomena result in the superior performance of frequency hopped systems at low vehicle speeds. Frequency hopping essentially eliminates the deleterious effects of the long fades common to the slow fading channel.

Performance Characterization

Another key set of statistics gathered on the residential route are the various bit-error-rates (BER) for both the information (decoded) bits and the channel bits. As indicated in Fig. 5, the error-correction coding methods employed provide valuable improvements in the BER performance. In this plot, the complete data sets for both the channel and decoded BERs are shown as a function of the signal-to-noise ratio (SNR) at the receiver. This data set was recorded on the downlink with both frequency hopping and antenna diversity enabled. General trends can be noted, such as the gains available through the use of the forward error correction coding, but the scatter plots of the complete data sets are not convenient for data comparison. Thus, a data reduction method is employed through the application of a "median filter," a process by which the data is grouped into signal strength bins in increments of 1 dB after which the median BER in each bin is determined. This filtering process provides a standard method for viewing and comparing the various field data sets. Fig. 5 also displays the median-filtered curve fit of the channel and decoded BER versus SNR for each of the scatter plot sets. Fig. 6 displays the median-filtered curve fits when hopping is either enabled or disabled, showing the significant gain available from frequency hopping. Both system configurations have antenna diversity enabled.

Generally speaking, given suitable error detection and mitigation, acceptable voice quality can be achieved at a 10^{-3} decoded BER. In the field data analysis, the SNR required to achieve this BER is termed the sensitivity point of the radio and has been determined for a variety of system configurations. The sensitivity point for each configuration is obtained from BER plots such

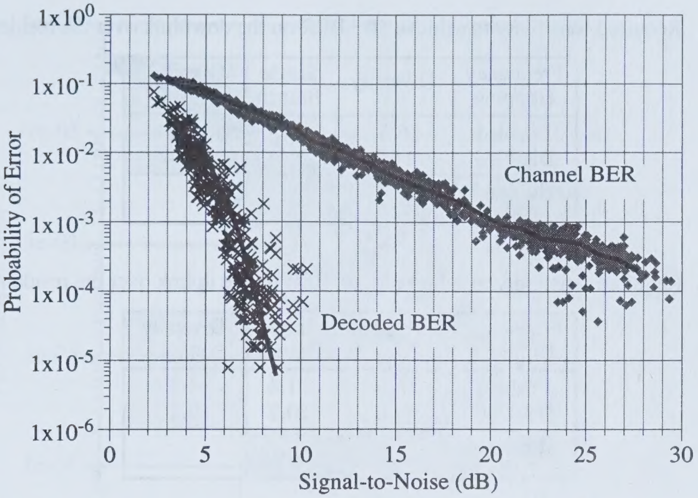


Fig. 5. Scatter plot of downlink field test BER data with median curve fits. Antenna diversity and frequency hopping are enabled.

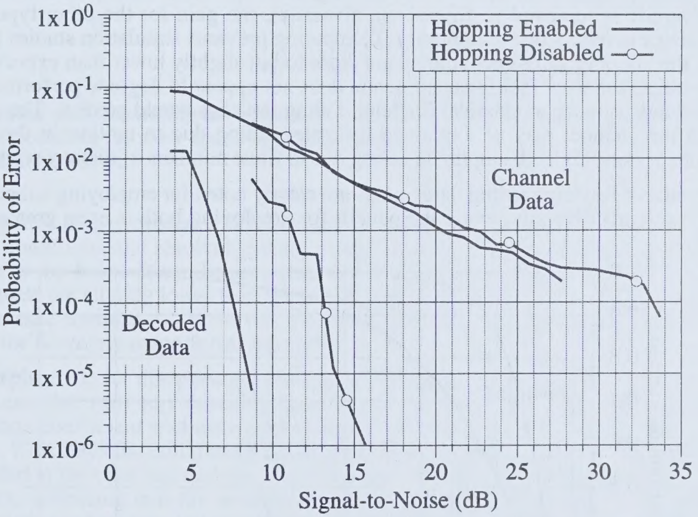


Fig. 6. Downlink BER median curve fits for both hopping configurations. Antenna diversity is enabled.

as that shown in Fig. 6. Table II summarizes the sensitivity of the SFH prototype configured with each of the four permutations of frequency hopping and antenna diversity from downlink data gathered in the residential neighborhood. Since the prototype system also operates as a duplex link, simultaneous analysis of both the uplink and downlink can be performed. Table III summarizes the sensitivity encountered for the four permutations on the uplink.

Tables II and III also summarize the antenna diversity gain and frequency hopping gain experienced over the possible combinations. With frequency hopping disabled, slow speed

Table II. Required sensitivity to achieve 10^{-3} BER on the downlink over the residential route.

Frequency Hopping	Diversity	Single Branch	Diversity Gain
Enabled	6.5	13.4	6.9
Disabled	11.2	21.6	10.4
Hopping Gain	4.7	8.2	

Table III. Required sensitivity to achieve a 10^{-3} BER on the uplink over the residential route.

Frequency Hopping	Diversity	Single Branch	Diversity Gain
Enabled	7.1	11.6	4.5
Disabled	11.1	20.2	9.1
Hopping Gain	4.0	8.6	

operation results in nearly a 5 dB degradation with diversity reception and over 8 dB degradation with single branch reception on the downlink. Note from the tables that as one type of diversity is removed (e.g., antenna diversity), the gain for the other type increases (e.g., frequency diversity due to hopping). Comparing previous simulation studies [1,3] to the field data, the hopping gains encountered are close to but slightly lower than expected. Based on these studies, the slow fading phenomenon does not appear to degrade performance of the non-hopped link as much as classical Rayleigh-fading statistics would predict. The diminished severity in the channel may be explained by faster fading due to motion in the scattering environment or non-Rayleigh amplitude fading due to finite numbers of distinct scatterers.

In the presence of Rayleigh fading, large gains are clearly noted for employing either frequency hopping or antenna diversity, while the benefit for employing both is even greater. Similar

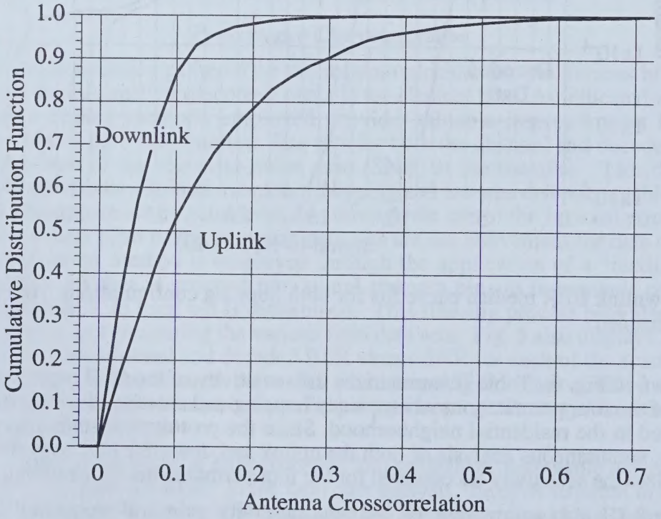


Fig. 7. CDFs of base and mobile antenna crosscorrelation coefficients.

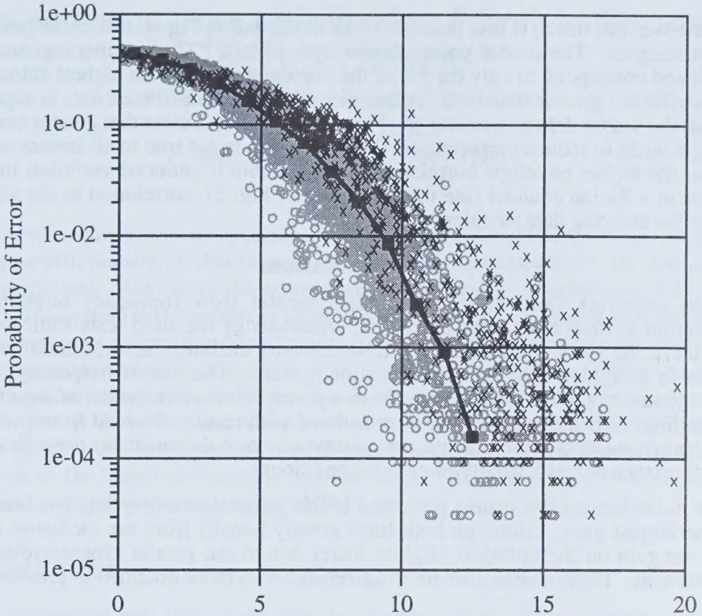


Fig. 8. A scatter plot illustrating the impact of envelope correlation on the probability of error after decoding for slow speed operation with frequency hopping enabled and antenna diversity disabled.

significant gains in performance were noted in [6] in an environment with multipath propagation conditions and resolvable delay spread, thus confirming the prototype's robust performance in the field. It can be seen that when antenna diversity is disabled, the uplink sensitivity points are slightly lower than those of the downlink (due to a lower implementation loss in the basesite equipment). However, when both frequency hopping and antenna diversity are enabled, the downlink outperforms the uplink.

A possible explanation for this phenomenon is a reduced diversity gain due to the lack of local scatterers near the roof-top basesite antennas. To investigate this hypothesis, the crosscorrelation coefficient was measured and recorded simultaneously at both the base and mobile. Fig. 7 displays the cumulative distribution function of the gathered crosscorrelation values recorded at the base and mobile. The large majority of crosscorrelation values are low for both links, indicating that the antenna diversity gain obtained will often be significant because of relative independence between the receiver branches. As the figure also indicates, the base exhibits higher crosscorrelation values than the mobile. With the increased correlation of the received signals at the base, a lower gain can be expected from the combining process, resulting in the lower antenna diversity gain values presented in Table III.

Thus, correlation between the received signals at each antenna port has been shown to reduce the spatial diversity gain. In a similar fashion, it can be expected that correlation in the envelope of the received signal at either antenna port may reduce the frequency hopping gain. To illustrate this, Fig. 8 plots the sample points from a field test where frequency hopping has been enabled but antenna diversity is disabled. This particular data set, captured on the uplink, is shown because the effects of correlation in the envelope on decoded BER are diminished by the presence of the complementary antenna diversity. In the figure, the scatter points are denoted with a gray circle when the envelope autocorrelation coefficient averaged over the span of the interleaver (40 msec) is less than 0.35. As illustrated in Fig. 4, this corresponds to 95% of the data samples. The scatter points shown with a black "X" are superimposed upon the gray points and correspond to only the 5% of the population having the highest autocorrelation, that is, a coefficient greater than 0.35. A line plot of the median-filtered data is superimposed on top of all the scatter data as a reference. From the figure, it is clear that greater correlation in the envelope tends to reduce performance. However, this is not true in all instances. In some cases when the higher envelope correlation is attributable to reduced variation in the faded envelope as in a Rician channel (see the histogram of Fig. 3), correlation in the signal is not responsible for outlying data samples.

IV. Conclusions

Uplink and downlink field data for an experimental slow frequency hopped personal communication system have been presented. Analysis of the field tests indicates that the combination of the space, frequency, and time diversity attributes incorporated into the PCS unit results in a highly reliable communication system. The use of frequency hopping in particular appears to provide excellent gains in a propagation environment of significant slow Rayleigh fading. When this capability is considered with results obtained from previous field tests in other propagation environments, the prototype system demonstrates great flexibility and robust performance across a variety of mobile conditions.

Within the noise-limited test results presented in this paper, space diversity has been found to provide the largest gain. Although both links greatly benefit from the inclusion of antenna diversity, the gain on the uplink is slightly lower due to the greater crosscorrelation of the basesite antennas. These results also are in agreement with those obtained in previous tests.

Acknowledgments

The authors gratefully acknowledge the efforts of Mark Cudak, Dave Gurney and Steve Kuffner, all of the Communication Systems Research Laboratory at Motorola, for the design and development of the research prototype. In addition, the authors would like to thank Marc Verity, Steve Kuffner, Andrew Bateman, Doug Dominiak, Mike Eberhardt and Colin Frank for their participation in gathering the field data.

References

[1] P. D. Rasky, G. M. Chiasson, D. E. Borth, and R. L. Peterson, "Slow Frequency-Hop TDMA/CDMA for Macrocellular Personal Communications," *IEEE Personal Communications Magazine*, pp. 26-35, Second Quarter, 1994.

[2] P. D. Rasky, G. M. Chiasson, and D. E. Borth, "An Experimental Slow Frequency-Hopped Personal Communication System for the Proposed U.S. 1850-1990 MHz Band," *Proceedings of the Second International Conference on Universal Personal Communications,* pp. 931-935, Ottawa, Canada, October 12-15, 1993.

[3] D. E. Borth, P. D. Rasky, G. M. Chiasson, and J. F. Kepler, "Frequency Hopped Systems for PCS," *Proceedings of the IEEE Third International Symposium on Spread Spectrum Techniques & Applications*, pp. 105-114, Oulu, Finland, July 4-6, 1994.

[4] W. C. Jakes (ed.), *Microwave Mobile Communications*, New York: John Wiley & Sons, 1974.

[5] P. D. Rasky, G. M. Chiasson and D. E. Borth, "Hybrid Slow Frequency-Hop/CDMA-TDMA as a Solution for High-Mobility, Wide-Area Personal Communications," *Proceedings of the Fourth Winlab Workshop on Third Generation Wireless Information Networks,* pp. 199-215, East Brunswick, New Jersey, October 19-20, 1993.

[6] G. M. Chiasson, J. F. Kepler, P. D. Rasky, and D. M. Tolli, "Field Measurements of a Prototype Slow Frequency Hop Personal Communication System," *Proceedings of the 1995 IEEE 45th Vehicular Technology Conference*, pp. 118-124, Rosemont, Illinois, July 26-28, 1995.

Limits on Performance with Random Polyphase Sequences in DS/SSMA Systems *

Tat M. Lok and James S. Lehnert
Purdue University
West Lafayette, IN 47907-1285

Abstract

Direct-sequence spread-spectrum multiple-access (DS/SSMA) communication systems with random m-phase sequences, for even m, are considered. By examining the asymptotic behavior of the normalized multiple-access interference (MAI), we find that a system with random m-phase sequences, for $m > 4$, should have the same performance as one with random quadriphase sequences asymptotically. However, a system with random m-phase sequences, for $m \geq 4$, may perform better than one with random binary sequences when the number of simultaneous users is relatively small. A new Gaussian approximation is proposed to estimate the probabilities of error in these systems. For the case of binary sequences, the new approximation reduces to the Improved Gaussian Approximation (IGA) which has been shown to provide close estimates to the probabilities of error.

1. Introduction

Polyphase sequences can have good correlation properties. Hence, there have been interest in using polyphase sequences in direct-sequence spread-spectrum multiple-access (DS/SSMA) communication systems, e.g., [1]-[3]. In most cases, the authors are interested in certain groups of sequences. The performance of the systems is evaluated under the deterministic sequence model, and the results apply only to those sequences investigated. The average performance of systems using polyphase sequences in general remains unknown. In order to investigate this problem, we adopt the random sequence model.

Compared to the deterministic sequence model, there are relatively few results with the random sequence model [4]-[8]. In [4]-[6], only the standard binary DS/SSMA communication system is considered. In [7] and [8], quadrature systems are also considered. However, in these systems, the data on the in-phase and the quadrature channels are assumed to be independent. Hence, the results do not cover the case of a system with binary data and random quadriphase sequences.

In this paper, we investigate the performance of systems with random m-phase sequences for even m. We consider the asymptotic behavior of the normalized multiple-access interference (MAI) and its various implications.

In Section 2, we describe the system model and derive an expression for the conditional MAI. In Section 3, we derive the joint characteristic function of the normalized MAI from different interferers. We consider the limit of the joint characteristic function as the number of chips N per symbol interval tends to infinity. We also consider the probability distribution function of the limiting MAI. In Section 4, we present a new Gaussian approximation, which is different from the Gaussian approximation [9] widely used in the literature. We refer to the latter as the Standard Gaussian Approximation (SGA). For the case of binary sequences, the new Gaussian approximation reduces to the Improved Gaussian Approximation (IGA) in [6]. It has been shown in [6] that the IGA provides better estimates than the SGA.

---

*This research was supported in part by the U.S. Army Research Office under grant DAAH04-95-I-0246.

2. System Model

In this section, we describe the model of the DS/SSMA systems. The kth user generates a data signal $b_k(t)$ that is given by

$$b_k(t) = \sum_{j=-\infty}^{\infty} b_j^{(k)} p_T(t - jT), \tag{1}$$

where T is the symbol duration, and the unit rectangular pulse $p_T(t)$ is given by $p_T(t) = 1$ for $0 \leq t < T$, and $p_T(t) = 0$, otherwise. The data symbols $b_j^{(k)}$ are independent and identically distributed (iid) random variables for which $\Pr\{b_j^{(k)} = 1\} = \Pr\{b_j^{(k)} = -1\} = 1/2$.

The kth user is given a randomly generated m-phase signature sequence where m is even throughout the paper. The kth user sequence is given by

$$a^{(k)} = (\ldots, a_0^{(k)}, a_1^{(k)}, \ldots, a_{N-1}^{(k)}, \ldots). \tag{2}$$

Each element $a_i^{(k)}$ is given by

$$a_i^{(k)} = \exp(j\phi_i^{(k)}), \tag{3}$$

where $\phi_i^{(k)}$ is modeled as a random variable uniformly distributed on the set $\{ 0, 2\pi/m, \ldots, 2(m-1)\pi/m \}$. We assume that the random variables $\phi_i^{(k)}$ and $\phi_{i'}^{(k)}$ are independent for $i \neq i'$. We also assume that they are independent of the data symbols $b_j^{(k)}$. The sequence $a^{(k)}$ is used to form the spectral spreading signal that is given by

$$a_k(t) = \sum_{i=-\infty}^{\infty} a_i^{(k)} \bar{\psi}\Big(\frac{t - iT_c}{T_c}\Big), \tag{4}$$

where the chip duration T_c is given by $T_c = T/N$, N is the number of chips per symbol interval, and $\bar{\psi}(t)$ is the common chip waveform for all signals. The chip waveform $\bar{\psi}(t)$ is real, time-limited to the interval $[0, 1)$, and normalized so that $\int_0^1 \bar{\psi}^2(t)dt = 1$.

The transmitted signal for the kth user is given by $\mathrm{Re}[\sqrt{2P} a_k(t) b_k(t) \exp(j\omega_c t)]$, where P is the power for the signal, and ω_c is the carrier frequency. Without loss of generality, we will consider the signal from the first user as the communicating signal and the signals from all other users as interfering signals throughout the paper.

The signal transmitted by the kth user is delayed by T_k, and the phase of this signal is shifted by $\tilde{\theta}_k$. Since we are are going to consider coherent detection, we can assume $T_1 = 0$ and $\tilde{\theta}_1 = 0$. We model each delay T_k, for $k \neq 1$, as a random variable that is uniformly distributed on the interval $[0, T_c)$ and independent of all other random variables in the system. We can make this assumption, instead of the usual assumption that T_k is uniformly distributed on $[0, T)$, because the statistics of $a_k(t)b_k(t)$ are the same as those of $a_k(t - iT_c)b_k(t - iT_c)$ for any integer i. We model each phase shift $\tilde{\theta}_k$, for $k \neq 1$, as a random variable that is uniformly distributed on the interval $[0, 2\pi)$ and independent of all other random variables in the system. We also assume that the communication channel is corrupted by additive white Gaussian noise (AWGN) with two-sided spectral density $N_0/2$.

The low-pass equivalent of the received signal is given by

$$r(t) = n(t) + \sum_{k=1}^{K} \sqrt{2P} a_k(t - T_k) b_k(t - T_k) \exp(j\theta_k), \tag{5}$$

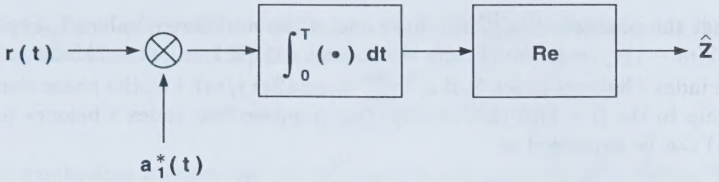


Figure 1: Receiver for the polyphase DS/SSMA system

where K is the number of simultaneous users, $n(t)$ is the low-pass equivalent of the AWGN, and $\theta_k = \tilde{\theta}_k - \omega_c T_k \pmod{2\pi}$ is again uniformly distributed on the interval $[0, 2\pi)$ and independent of T_k. We can assume that

$$r(t) = n(t) + \sqrt{2P} a_1(t) b_1(t) + \sum_{k=2}^{K} \sqrt{2P} a_k(t - T_k) \exp(j\theta_k) \tag{6}$$

since the statistics of $a_k(t)b_k(t)$ are the same as those of $a_k(t)$. This remains true when $b_i^{(k)} = \exp(j\varphi_i^{(k)})$, where $\varphi_i^{(k)}$ is uniformly distributed on some given subset of the set $\{$ 0, $2\pi/m$, …, $2(m-1)\pi/m$ $\}$. Hence, all subsequent results concerning the statistics of the MAI can be generalized to the case of polyphase data.

The decision statistic $\hat{Z}$ shown in Fig. 1 is given by

$$\hat{Z} = \hat{\eta} + \sqrt{2P} T b_0^{(1)} + \sqrt{2P} T_c \sum_{k=2}^{K} \hat{I}_k, \tag{7}$$

where

$$\hat{I}_k = \mathrm{Re}\left[\sum_{i=0}^{N-1} a_i^{(1)*} [a_{i-1}^{(k)} R_{\bar{\psi}}(T'_k) + a_i^{(k)} \hat{R}_{\bar{\psi}}(T'_k)] \exp(j\theta_k) \right] \tag{8}$$

is the MAI from the kth user. The random variable $\hat{\eta}$ accounts for the effect of the AWGN and is Gaussian with mean equal to zero and variance equal to $N_0 T$. The functions $R_{\bar{\psi}}(\cdot)$ and $\hat{R}_{\bar{\psi}}(\cdot)$ are the partial autocorrelation functions of the chip waveform defined similarly as in [4], for $0 \leq \tau < 1$, by $R_{\bar{\psi}}(\tau) = \int_0^\tau \bar{\psi}(t)\bar{\psi}(t+1-\tau)dt$ and $\hat{R}_{\bar{\psi}}(\tau) = \int_\tau^1 \bar{\psi}(t)\bar{\psi}(t-\tau)dt$, and $T'_k = T_k/T_c$. Since T_k is uniformly distributed on $[0, T_c)$, T'_k is uniformly distributed on $[0, 1)$. Regrouping terms, we can express (8) as

$$\begin{aligned} \hat{I}_k = \mathrm{Re}\Big[\; & \Big\{ a_{-1}^{(k)} a_0^{(1)*} R_{\bar{\psi}}(T'_k) + a_{N-1}^{(k)} a_{N-1}^{(1)*} \hat{R}_{\bar{\psi}}(T'_k) \\ & + \sum_{i=0}^{N-2} a_i^{(k)} a_i^{(1)*} [\hat{R}_{\bar{\psi}}(T'_k) + a_i^{(1)} a_{i+1}^{(1)*} R_{\bar{\psi}}(T'_k)] \Big\} \exp(j\theta_k) \; \Big]. \end{aligned} \tag{9}$$

We condition on $\phi^{(1)} = (\phi_0^{(1)}, \ldots, \phi_{N-1}^{(1)})$ and define the random variables $\zeta_i^{(k)}$ by

$$\zeta_i^{(k)} = \begin{cases} \phi_i^{(k)} - \phi_i^{(1)} \bmod 2\pi & \text{for } \; 0 \leq i \leq N-1, \\ \phi_{-1}^{(k)} - \phi_0^{(1)} \bmod 2\pi & \text{for } \; i = -1. \end{cases} \tag{10}$$

The random variables $\zeta_i^{(k)}$ are independent and uniformly distributed on the set $\{$ 0, $2\pi/m$, …, $2(m-1)\pi/m$ $\}$. We define $c_i^{(k)}$ by $c_i^{(k)} = \exp(j\zeta_i^{(k)})$. Then (9) can be rewritten as

$$\begin{aligned} \hat{I}_k = \mathrm{Re}\Big[\; & \Big\{ c_{-1}^{(k)} R_{\bar{\psi}}(T'_k) + c_{N-1}^{(k)} \hat{R}_{\bar{\psi}}(T'_k) \\ & + \sum_{i=0}^{N-2} c_i^{(k)} [\hat{R}_{\bar{\psi}}(T'_k) + a_i^{(1)} a_{i+1}^{(1)*} R_{\bar{\psi}}(T'_k)] \Big\} \exp(j\theta_k) \; \Big]. \end{aligned} \tag{11}$$

Notice that the product $a_i^{(1)}a_{i+1}^{(1)*}$ can have one of the m different values 1, $\exp(2\pi j/m)$, …, $\exp(2(m-1)\pi j/m)$. We classify the indices $i \in \{0, 1, \ldots, N-2\}$ into m different sets. The index i belongs to set S_q if $a_i^{(1)}a_{i+1}^{(1)*} = \exp(2q\pi j/m)$, i.e., the phase change from the ith chip to the $(i+1)$th chip is $-2q\pi/m$. Suppose that index i belongs to set S_{q_i}. Then (11) can be expressed as

$$\begin{aligned}\hat{I}_k = \; & R_{\bar{\psi}}(T_k')\cos(\zeta_{-1}^{(k)} + \theta_k) + \hat{R}_{\bar{\psi}}(T_k')\cos(\zeta_{N-1}^{(k)} + \theta_k) \\ & + \sum_{i=0}^{N-2}\left[\hat{R}_{\bar{\psi}}(T_k')\cos(\zeta_i^{(k)} + \theta_k) + R_{\bar{\psi}}(T_k')\cos(\zeta_i^{(k)} + \theta_k + \tfrac{2q_i\pi}{m})\right].\end{aligned} \tag{12}$$

3. Asymptotic Properties of MAI

We normalize the decision statistic $\hat{Z}$ by $1/(\sqrt{2NPT_c})$. The result is given by

$$Z = \sqrt{N}b_0^{(1)} + I + \eta, \tag{13}$$

where the normalized overall MAI $I = \sum_{k=2}^{K} I_k$, the normalized MAI from the kth user $I_k = \hat{I}_k/\sqrt{N}$, and η is a Gaussian random variable with mean equal to zero and variance equal to $NN_0/(2E_b)$.

We find the joint characteristic function of the normalized MAI from the interferers as follows: If we condition on $\phi^{(1)}$, the random variables I_k, for $2 \le k \le K$, become iid. Hence, we first find the common characteristic function of I_k conditioned on $\phi^{(1)}$. The characteristic function of I_k, conditioned on $\phi^{(1)}$, T_k', and θ_k, is given by

$$\begin{aligned}\Phi_{I_k}(u; \phi^{(1)}, T_k', \theta_k) & = \mathbf{E}[\exp(juI_k)|\phi^{(1)}, T_k', \theta_k] \\ & = \Phi_{I_*}(u; \phi^{(1)}, T_k', \theta_k)\end{aligned} \tag{14}$$

where

$$\Phi_{I_*}(u; \phi^{(1)}, \tau, \theta) = \beta(u, \tau, \theta)\gamma(u, \tau, \theta)\prod_{q=0}^{m-1}\alpha_q(u, \tau, \theta)^{|S_q|}, \tag{15}$$

$$\beta(u, \tau, \theta) = \frac{2}{m}\sum_{n=0}^{m/2-1}\cos\left(uN^{-\frac{1}{2}}R_{\bar{\psi}}(\tau)\cos(\tfrac{2n\pi}{m} + \theta)\right), \tag{16}$$

$$\gamma(u, \tau, \theta) = \frac{2}{m}\sum_{n=0}^{m/2-1}\cos\left(uN^{-\frac{1}{2}}\hat{R}_{\bar{\psi}}(\tau)\cos(\tfrac{2n\pi}{m} + \theta)\right), \tag{17}$$

$$\alpha_q(u, \tau, \theta) = \frac{2}{m}\sum_{n=0}^{m/2-1}\cos\left(uN^{-\frac{1}{2}}[\hat{R}_{\bar{\psi}}(\tau)\cos(\tfrac{2n\pi}{m} + \theta) + R_{\bar{\psi}}(\tau)\cos(\tfrac{2n\pi}{m} + \theta + \tfrac{2q\pi}{m})]\right), \tag{18}$$

and $|S_q|$ is the cardinality of the set S_q. The characteristic function of I_k, conditioned on $\phi^{(1)}$ only, is given by

$$\Phi_{I_k}(u; \phi^{(1)}) = \frac{1}{2\pi}\int_0^1\int_0^{2\pi}\Phi_{I_*}(u; \phi^{(1)}, T_k', \theta_k)\,d\theta_k\,dT_k'. \tag{19}$$

Notice that the characteristic function depends on $\phi^{(1)}$ only through the cardinalities of the sets S_q. The joint characteristic function of the normalized MAI, conditioned on $\phi^{(1)}$, is given by

$$\begin{aligned}\Phi_{\mathbf{I}}(\mathbf{u}; \phi^{(1)}) & = \mathbf{E}[\exp(j\mathbf{u}\cdot\mathbf{I})|\phi^{(1)}] \\ & = \prod_{k=2}^{K}\Phi_{I_k}(u_k; \phi^{(1)}),\end{aligned} \tag{20}$$

where $\mathbf{I} = (I_2, I_3, \ldots, I_K)$, and $\mathbf{u} = (u_2, u_3, \ldots, u_K)$. The unconditional joint characteristic function of the normalized MAI is given by

$$\Phi_{\mathbf{I}}(\mathbf{u}) = \frac{1}{m^N} \sum \Phi_{\mathbf{I}}(\mathbf{u}; \phi^{(1)}), \tag{21}$$

where the summation is taken over all m^N possible sequences for $\phi^{(1)}$. Notice from (15), (19) and (20) that if two different sequences of $\phi^{(1)}$ have the same cardinalities for the sets S_q, then the conditional joint characteristic function in (20) would be the same. Given a set of non-negative integers q_j, $0 \leq j \leq m-1$, with $\sum_{j=0}^{m-1} q_j = N-1$, the number of sequences $\phi^{(1)}$ with cardinalities $|S_j| = q_j$ is given by

$$m \frac{(N-1)!}{q_0! q_1! \ldots q_{m-1}!}. \tag{22}$$

Hence, (21) can be expressed as

$$\Phi_{\mathbf{I}}(\mathbf{u}) = \frac{1}{m^{N-1}} \sum_{q_0+\ldots+q_{m-1}=N-1} \left[\frac{(N-1)!}{q_0! q_1! \ldots q_{m-1}!} \Phi_{\mathbf{I}}(\mathbf{u}; q_0, \ldots, q_{m-1}) \right], \tag{23}$$

where the summation is now over all possible choices of q_j's such that $q_0 + \ldots + q_{m-1} = N-1$, and $\Phi_{\mathbf{I}}(\mathbf{u}; q_0, \ldots, q_{m-1})$ denotes the conditional joint characteristic function of the normalized MAI given any sequence with cardinalities $|S_0| = q_0, \ldots, |S_{m-1}| = q_{m-1}$.

We consider the limit of the joint characteristic function in (23) as N tends to infinity. We first consider the case for $K = 2$, i.e., the limit of the characteristic function with only one interfering signal. The characteristic function of I_2 can be rewritten as

$$\begin{aligned} \Phi_{I_2}(u) &= \frac{1}{2\pi} \int_0^1 \int_0^{2\pi} \beta(u, T_2', \theta_2) \gamma(u, T_2', \theta_2) \\ &\quad \frac{1}{m^{N-1}} \sum_{q_0+\ldots+q_{m-1}=N-1} \left[\frac{(N-1)!}{q_0! q_1! \ldots q_{m-1}!} \prod_{j=0}^{m-1} \alpha_j(u, T_2', \theta_2)^{q_j} \right] d\theta_2 dT_2' \\ &= \frac{1}{2\pi} \int_0^1 \int_0^{2\pi} \beta(u, T_2', \theta_2) \gamma(u, T_2', \theta_2) \left[\frac{1}{m} \sum_{j=0}^{m-1} \alpha_j(u, T_2', \theta_2) \right]^{N-1} d\theta_2 dT_2'. \end{aligned} \tag{24}$$

From (16)-(18), we see that the absolute value of the integrand in (24) is bounded by 1. Hence, by the Dominated Convergence Theorem, we can find the limit of $\Phi_{I_2}(u)$ as N tends to infinity by finding the integral of the limit of the integrand in (24), provided that the limit of the integrand exists a.e. We first consider, for sufficiently large N, the natural logarithm of the integrand

$$g_N(u; T_2', \theta_2) = \log[\beta(u, T_2', \theta_2)] + \log[\gamma(u, T_2', \theta_2)] + (N-1) \log \left[\frac{1}{m} \sum_{j=0}^{m-1} \alpha_j(u, T_2', \theta_2) \right]. \tag{25}$$

Applying L'Hospital's rule, we obtain the limit given by

$$\lim_{N \to \infty} g_N(u; T_2', \theta_2) = -u^2 V_2 / 2, \tag{26}$$

where

$$V_2 = \frac{2}{m^2} \sum_{q=0}^{m-1} \sum_{n=0}^{m/2-1} \left[\hat{R}_{\bar{\psi}}(T_2') \cos\left(\frac{2n\pi}{m} + \theta_2 \right) + R_{\bar{\psi}}(T_2') \cos\left(\frac{2n\pi}{m} + \theta_2 + \frac{2q\pi}{m} \right) \right]^2. \tag{27}$$

Equation (27) can be simplified to

$$V_2 = \begin{cases} [\hat{R}^2_{\bar{\psi}}(T_2') + R^2_{\bar{\psi}}(T_2')]\cos^2(\theta_2) & \text{for} \quad m = 2, \\ [\hat{R}^2_{\bar{\psi}}(T_2') + R^2_{\bar{\psi}}(T_2')]/2 & \text{for} \quad m \geq 4. \end{cases} \tag{28}$$

Since the exponential function is continuous, the limit of the integrand of the last expression in (24) is $\exp(-u^2V_2/2)$. Hence, we have

$$\lim_{N\to\infty} \Phi_{I_2}(u) = \frac{1}{2\pi}\int_0^1\int_0^{2\pi} \exp(-u^2V_2/2)d\theta_2 dT_2'. \tag{29}$$

This expression has the interpretation that, in the limit, the normalized MAI is Gaussian with mean equal to zero and variance equal to V_2 conditioned on the relative delay T_2' and the phase shift θ_2. (We interpret a Gaussian random variable with zero mean and zero variance as the constant zero.)

For $K = 3$, (23) becomes

$$\begin{aligned} \Phi_{\mathbf{I}}(\mathbf{u}) = \; & \Big(\frac{1}{2\pi}\Big)^2 \int_0^1\int_0^1\int_0^{2\pi}\int_0^{2\pi} \beta(u_2, T_2', \theta_2)\beta(u_3, T_3', \theta_3)\gamma(u_2, T_2', \theta_2)\gamma(u_3, T_3', \theta_3) \\ & \frac{1}{m^{N-1}} \sum_{q_0+\ldots+q_{m-1}=N-1} \left[\frac{(N-1)!}{q_0!q_1!\ldots q_{m-1}!} \prod_{j=0}^{m-1} \alpha_j(u_2, T_2', \theta_2)^{q_j}\alpha_j(u_3, T_3', \theta_3)^{q_j}\right] \\ & d\theta_2 d\theta_3 dT_2' dT_3'. \end{aligned} \tag{30}$$

We can still reduce the sum in the integrand to a multinomial. By similar arguments as before, we find the limit of the joint characteristic function by finding the limit of the integrand. The result is given by

$$\lim_{N\to\infty} \Phi_{\mathbf{I}}(\mathbf{u}) = \Big(\frac{1}{2\pi}\Big)^2 \int_0^1\int_0^1\int_0^{2\pi}\int_0^{2\pi} \exp\Big(-\frac{u_2^2V_2 + u_3^2V_3}{2}\Big) d\theta_2 d\theta_3 dT_2' dT_3', \tag{31}$$

where

$$V_k = \begin{cases} [\hat{R}^2_{\bar{\psi}}(T_k') + R^2_{\bar{\psi}}(T_k')]\cos^2(\theta_k) & \text{for} \quad m = 2, \\ [\hat{R}^2_{\bar{\psi}}(T_k') + R^2_{\bar{\psi}}(T_k')]/2 & \text{for} \quad m \geq 4. \end{cases} \tag{32}$$

Equation (31) has the interpretation that, in the limit, the normalized MAI are jointly Gaussian conditioned on the relative delays T_2' and T_3', and the phase shifts θ_2 and θ_3. Notice that (31) can be rewritten as

$$\lim_{N\to\infty} \Phi_{\mathbf{I}}(\mathbf{u}) = \Big[\lim_{N\to\infty} \Phi_{I_2}(u_2)\Big]\Big[\lim_{N\to\infty} \Phi_{I_3}(u_3)\Big]. \tag{33}$$

Hence, in the limit, the MAI from different interferers are independent. The generalization for larger values of K follows similarly, and we have the following main result.

Proposition 1 : For any finite $K > 1$, we have

$$\lim_{N\to\infty} \Phi_{\mathbf{I}}(\mathbf{u}) = \Big(\frac{1}{2\pi}\Big)^{K-1} \int_0^1\cdots\int_0^1\int_0^{2\pi}\cdots\int_0^{2\pi} \exp\Big(-\sum_{k=2}^{K}\frac{u_k^2V_k}{2}\Big) d\theta_2\ldots d\theta_K \, dT_2'\ldots dT_K', \tag{34}$$

where V_k is given by (32). In the limit, the normalized MAI from different interferers are independent, and they are jointly Gaussian conditioned on the set $S_{\tau,\theta} = \{T_k', \theta_k : k = 2, 3, \ldots, K\}$.

By setting $u_2 = u_3 = \ldots = u_K = u$, we obtain the characteristic function of the normalized overall MAI and the following corollary.

Corollary 1 : Conditioned on the set $S_{\tau,\theta}$, the normalized overall MAI I tends to a Gaussian random variable with mean equal to zero and variance equal to $V = \sum_{k=2}^{K} V_k$.

From (32), we see that, for $m \geq 4$, the variance depends only on the set $S_{\tau,\theta}$, but not on m. As a result, the characteristic function, and hence the probability distribution function, of the normalized overall MAI does not depend on m in the limit, provided $m \geq 4$. Hence, we have the following corollary.

Corollary 2 : The average performance of a system with random m-phase sequences, for $m > 4$, is asymptotically the same as one with random quadriphase sequences.

The distribution function of the limiting normalized overall MAI is given by

$$F(x) = \Big(\frac{1}{2\pi}\Big)^{K-1} \int_0^1 \ldots \int_0^1 \int_0^{2\pi} \ldots \int_0^{2\pi} F(x; V) d\theta_2 \ldots d\theta_K dT'_2 \ldots dT'_K, \tag{35}$$

where

$$F(x; V) = \begin{cases} \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi V}} \exp\Big(-\frac{y^2}{2V}\Big) dy & \text{if } V > 0, \\ U(x) & \text{if } V = 0, \end{cases} \tag{36}$$

with $U(x) = 0$ for $x < 0$ and $U(x) = 1$ for $x \geq 0$, and $V = \sum_{k=2}^{K} V_k$. This distribution function is the same for all even m except $m = 2$. Hence we are interested in comparing the case for $m = 2$, which we denote by $F_B(x)$, and the case for $m \geq 4$, which we denote by $F_Q(x)$. It can be shown that both distribution functions correspond to random variables with mean equal to zero and variance equal to

$$\mathbf{E}[V] = (K-1)\mathbf{E}[\hat{R}^2_{\bar{\psi}}(T'_k) + R^2_{\bar{\psi}}(T'_k)]/2 = (K-1)M_{\bar{\psi}}, \tag{37}$$

where $M_{\bar{\psi}}$ is defined similarly as in [10] by $M_{\bar{\psi}} = \mathbf{E}[R^2_{\bar{\psi}}(T'_k)] = \mathbf{E}[\hat{R}^2_{\bar{\psi}}(T'_k)]$. However, a random variable corresponding to $F_B(x)$ takes on large values with higher probabilities as given in the following proposition.

Proposition 2 : For $x > \sqrt{3(K-1)}$, $1 - F_B(x) > 1 - F_Q(x)$.

Proof: See [11].

Proposition 2 could have the following implication. Suppose that for some sufficiently large but finite N, we can closely approximate the distribution function of the normalized MAI by the distribution function of the limiting normalized MAI. Suppose also that thermal noise is negligible so that all errors are due to MAI. The absolute value of the normalized signal contribution is $\sqrt{N}$. Hence, the probability of error is $1 - F(\sqrt{N})$. If $N > 3(K-1)$, Proposition 2 implies that the average probability of error in a system with binary sequences is larger than that in a system with m-phase sequences for $m \geq 4$. In other words, if the number of simultaneous users is relatively small ($K < N/3 + 1$), a system with polyphase sequences should perform better.

4. A Gaussian Approximation

From Corollary 1, we know that if we condition on the set $S_{\tau,\theta}$, the normalized overall MAI tends to a Gaussian random variable. Hence, we may approximate the conditional MAI by a Gaussian random variable for large N. When the conditional MAI is approximated by a Gaussian random variable, the approximate average probability of error

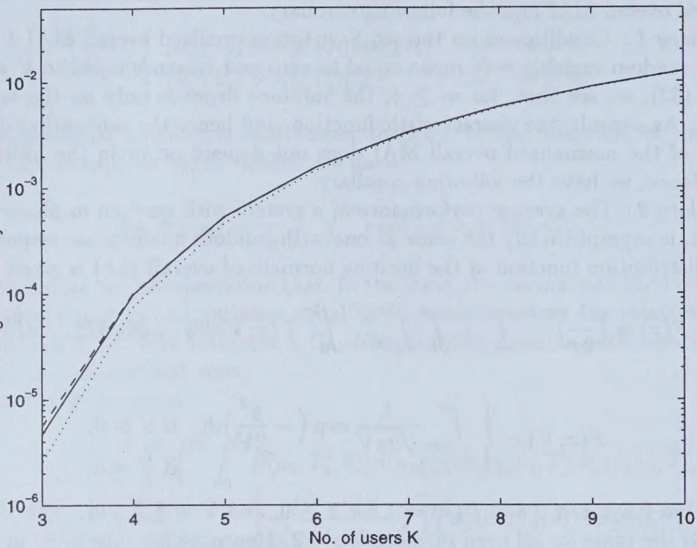


Figure 2: Average probabilities of error for a QPSK system (N=31, solid line: true value, broken line: new approximation, dotted line: SGA)

$\hat{P}_{e|S_{\tau,\theta}}$ conditioned on the set $S_{\tau,\theta}$ is given by

$$\hat{P}_{e|S_{\tau,\theta}} = Q\left(\sqrt{\frac{N}{V + NN_0/(2E_b)}}\right). \tag{38}$$

It depends on the set $S_{\tau,\theta}$ only through V. The unconditional approximate average probability of error $\hat{P}_e$ is given by

$$\hat{P}_e = \mathbf{E}\left[Q\left(\sqrt{\frac{N}{V + NN_0/(2E_b)}}\right)\right]. \tag{39}$$

To apply this approximation, we do not need to evaluate multiple integrals. Instead, we only need to find the probability distribution function of V. Details of carrying out this approximation can be found in [6] and [7], in which similar algorithms have been derived for the standard binary system and some quadrature systems.

In general, it is difficult to check the accuracy of the new approximation. However, in some special cases, we are able to verify the accuracy of our approximation. For the case of binary sequences, this approximation reduces to the IGA in [6]. It has been verified in [6] that the IGA provides better estimates to the probabilities of error than the SGA even for relatively small N. For the case of quadriphase data and quadriphase sequences, the results from the new approximation are shown in Fig. 2, along with the true probabilities of error and the results from the SGA. It can be seen that the new approximation provides better estimates than the SGA.

We would like to further compare the new approximation with the SGA. According to the standard Gaussian approximation, a system with random m-phase sequences, for $m \geq 4$, should have the same performance as one with random binary sequences, provided that N and K are the same in both systems. However, according to the new approximation, a system with random m-phase sequences should perform better if the number of simultaneous users is relatively small as shown in the following proposition.

Proposition 3 : Let $V_B(S_{\tau,\theta})$ be the variance of the conditional limiting MAI given the set $S_{\tau,\theta}$ in a system with random binary sequences. Let $V_Q(S_{\tau,\theta})$ be the counterpart in a system with random m-phase sequences, for $m \geq 4$. If

$$K - 1 < \Big(1 - \frac{3N_0}{2E_b}\Big)\frac{N}{3}, \tag{40}$$

then

$$\mathbf{E}\Bigg[Q\Bigg(\sqrt{\frac{N}{V_B + \frac{NN_0}{2E_b}}}\Bigg)\Bigg] > \mathbf{E}\Bigg[Q\Bigg(\sqrt{\frac{N}{V_Q + \frac{NN_0}{2E_b}}}\Bigg)\Bigg]. \tag{41}$$

Proof : See [11].

However, when the number of simultaneous users is large. The following proposition holds. It says that as both N and K get large, the probabilities of error predicted by the new approximation converge to those values predicted by the SGA.

Proposition 4 : Let the ratio $\rho = (K-1)/N$ be fixed. Then

$$\lim_{K\to\infty} \mathbf{E}\Bigg[Q\Bigg(\sqrt{\frac{N}{V + \frac{NN_0}{2E_b}}}\Bigg)\Bigg] = Q\Bigg(\sqrt{\frac{1}{\rho M_{\bar{\psi}} + \frac{N_0}{2E_b}}}\Bigg). \tag{42}$$

Proof : Let

$$C(x) = Q(1/\sqrt{\rho x + N_0/(2E_b)}). \tag{43}$$

Then the left side of (42) is just the limit of $\mathbf{E}[C(\bar{V})]$ where $\bar{V} = V/(K-1)$ is the arithmetic mean of the $K-1$ iid random variables V_k. By the Strong Law of Large Numbers, $\bar{V}$ tends to $\mathbf{E}[V_k] = M_{\bar{\psi}}$ with probability one. Since $C(x)$ is a continuous function of x, $C(\bar{V})$ tends to $C(M_{\bar{\psi}})$ with probability one. Since $C(x)$ is bounded by 1, by the Dominated Convergence Theorem, we have (42). □

Suppose that we fix $\rho < [1 - 3N_0/(2E_b)]/3$. Then the new approximation predicts that a system with m-phase sequences, for $m \geq 4$, should perform better than one with binary sequences (Proposition 3). However, as both N and K get large, the probabilities of error for both systems converge to the same value as predicted by the SGA (Proposition 4), and, hence, the gain in performance through polyphase sequences diminishes and eventually disappears.

5. Conclusions

By considering the characteristic functions of the MAI, we find that, in the limit, the normalized MAI from different interferers are unconditionally independent and are jointly Gaussian conditioned on the relative delays and phases. We also find that a system with random m-phase sequences, for $m > 4$, should have the same performance as one with random quadriphase sequences asymptotically. We predict that a system with random m-phase sequences, for $m \geq 4$, may perform better than one with random binary sequences when the number of simultaneous users is relatively small. We have proposed a new Gaussian approximation to estimate the probabilities of error in these systems. According to the new approximation, a system with random m-phase sequences, for $m \geq 4$, should perform better than one with random binary sequences when the number of simultaneous users is relatively small. However, as the number of users increases, the gain diminishes.

References

[1] T. Li, M. Hatori and N. Suehiro, "Analysis of asynchronous direct-sequence spread-spectrum multiple-access communications with polyphase sequences," in *Proc. IEEE GLOBECOM*, Nov. 1987, vol. 1, pp. 16.6.1-16.6.5.

[2] A. W. Lam and F. M. Ozluturk, "Performance bounds for DS/SSMA communications with complex signature sequences,"*IEEE Trans. Commun.*, vol. COM-40, pp. 1607-1614, Oct. 1992.

[3] H. Fukumasa, R. Kohno and H. Imai, "Design of psuedonoise sequences with good odd and even correlation properties for DS/CDMA,"*IEEE Journal on Selected Areas in Commun.*, vol. 12, pp. 828-836, Jun. 1994

[4] J. S. Lehnert and M. B. Pursley, "Error probabilities for binary direct-sequence spread-spectrum communications with random signature sequences,"*IEEE Trans. Commun.*, vol. COM-35, pp. 87-98, Jan. 1987.

[5] R. K. Morrow, Jr., and J. S. Lehnert, "Bit-to-bit error dependence in slotted DS/SSMA packet systems with random signature sequences,"*IEEE Trans. Commun.*, vol. COM-37, pp. 1052-1061, Oct. 1989.

[6] R. K. Morrow, Jr., and J. S. Lehnert, "Packet throughput in slotted ALOHA DS/SSMA radio systems with random signature sequences,"*IEEE Trans. Commun.*, vol. COM-40, pp. 1223-1230, Jul. 1992.

[7] T. M. Lok and J. S. Lehnert, "Error probabilities for generalized quadriphase DS/SSMA communication systems with random signature sequences," to appear in *IEEE Trans. Commun.*

[8] E. Geraniotis and B. Ghaffari, "Performance of binary and quaternary direct-sequence spread-spectrum multiple-access systems with random signature sequences,"*IEEE Trans. Commun.*, vol. COM-39, pp. 713-724, May 1991.

[9] M. B. Pursley, "Performance evaluation for phase-coded spread-spectrum multiple-access communication - Part I: System analysis,"*IEEE Trans. Commun.*, vol. COM-25, pp. 795-799, Aug. 1977.

[10] M. B. Pursley, F. D. Garber, and J. S. Lehnert, "Analysis of generalized quadriphase spread-spectrum communications," in *Conf. Rec., IEEE Int. Conf. Commun.*, Jun. 1980, vol. 1, pp. 15.3.1-15.3.6.

[11] T. M. Lok and J. S. Lehnert, "An asymptotic analysis of DS/SSMA communication systems with random polyphase signature sequences," to appear in *IEEE Trans. Inform. Theory*.

Throughput Optimization in Faded Multicarrier Systems

Dennis L. Goeckel[1] and Wayne E. Stark[1]
EECS Department
University of Michigan
Ann Arbor, MI 48109

Abstract

In this work, a multiple-access multicarrier system is presented with a concatenated coding scheme that encompasses most of the systems presented in the literature - a simple soft-decision decoded inner code and a complicated hard-decision decoded outer code. A measure for the quality of a system is then defined as the total number of bits transmitted reliably across the channel. The joint optimization of this throughput over the rate of the inner code and the number of users is considered for a coherent BPSK system, and tight bounds on the optimal thoughput are presented.

1 Introduction

The cellular phone industry has undergone explosive growth in recent years, and the capacity of digital cellular cellular systems is leading to the rapid introduction of digital cellular systems to replace their analog counterparts. Because of the expense of bandwidth in this new market, a digital communications protocol that accommodates as large a number of users as possible in a given bandwidth is required. The nature of the channel also demands a system that is able to deal with multipath fading of the signal. One solution to this problem has been direct-sequence code-division-multiple-access (DS-CDMA), whose spreading of the users' signals across a wide bandwidth offers not only mitigation of the multi-user interference, but also good time resolution that allows for multipath combining in the form of a rake receiver when DS-CDMA is used over a frequency selective fading channel.

---

[1] This work was supported in part by a National Science Foundation Graduate Fellowship, a Rackham Pre-Doctoral Fellowship, the National Science Foundation under grant NCR-9115969, and the Army Research Office under contract number DAAH04-95-I-0246.

While DS-CDMA uses a wide bandwidth to allow it to take advantage of path diversity through the use of a rake receiver, another method of achieving diversity with a large bandwidth is to transmit multiple narrowband signals. While the entire bandwidth of the system may exceed the coherence bandwidth of the channel (and thus lead to frequency selective fading), each narrowband signal can be chosen to have a bandwidth much less than the coherence frequency of the channel and lead to nonselective fading. This alternative is achieving great popularity in Europe for digital broadcast applications. The classical method of producing the narrowband signals using the discrete Fourier transform (DFT) [6] will be referred to as orthogonal frequency division multiplexing (OFDM). OFDM sends a block of symbols at a time by using the DFT to form the transmitted signal, thus obviating the need to synchronize many narrowband carriers at the receiver. Note that OFDM implicitly assumes that the narrowband frequency regions are contiguous in frequency, which may lead to large correlation between adjacent frequency slots. Interleaving in time or frequency spreads this correlation out across separate codewords. To avoid confusion, the system presented in what follows will be denoted multicarrier CDMA (MC-CDMA) and expressed as bits modulating independent narrowband carriers. However, it should be remembered that for physical implementation, one should interleave carefully and take advantage of OFDM's implementation advantages.

Section 2 contains the formal definition of the concatenated coded multicarrier system that is considered. Section 3 contains the calculation of the probability of error for the inner code, which is subsequently used for throughput optimization in Section 4. Section 5 presents conclusions and continuing work on the project.

2 MC-CDMA System Definition

The channel consists of a large number of non-interfering narrowband channels, each subject to nonselective Rayleigh fading. Although it would be possible to put one user per channel and thus maximize the number of non-interfering users, it may be advisable on a fading channel to allow each user to use $L > 1$ channels to achieve frequency diversity. Thus it is assumed that the N channels are divided into q slots of L frequency channels each, $N = qL$. It is assumed that sufficient interleaving exists such that the L frequency channels in a given slot have independent fading variables. This allows not only for analytic simplicity (the correlated case is difficult) but also provides the best obtainable performance. While allowing one user per slot eliminates multi-user interference, multi-user capacity arguments suggest that restricting each user to using disjoint resources does not achieve system capacity. For this reason, each slot will be allowed to carry $K > 1$ users, each signalling over all L channels in the slot, thereby achieving L^{th} order diversity. Figure 1 displays a picture of the set-up.

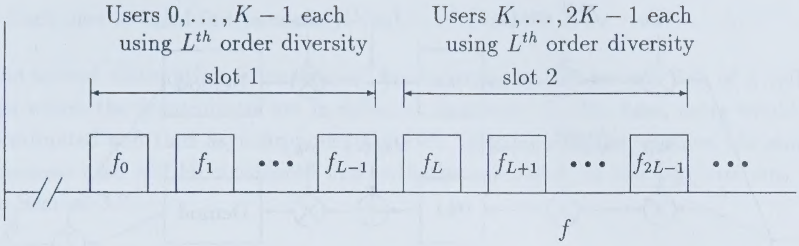

Users $0,\ldots,K-1$ each using L^{th} order diversity
slot 1
Users $K,\ldots,2K-1$ each using L^{th} order diversity
slot 2
f_0
f_1
f_{L-1}
f_L
f_{L+1}
f_{2L-1}
f

Figure 1: Frequency Channel Assignment to Slots

Although user separation in a given slot could be achieved by time spreading on each channel, this would involve bandwidth expansion of each of the frequency channels, which we would like to avoid. Instead, a block of n user data bits is encoded into L channel bits and the codeword $\{a_{k,l}, l = 0, \ldots, L-1\}$ of the k^{th} user is sent across the L channels that the user occupies. If the code used is a rate $1/L$ repetition code, this process is a simple spreading operation and the codeword is the spreading code. Since this is the main problem that has been explored to this point, it will be assumed that repetition coding is being used from here forward. At the receiver, despreading is done for the k^{th} user by multiplying the received signal off of the l^{th} channel by $a_{k,l}$ before combining. Thus, spreading is done for each user *across* the L channels in a user's slot.

A block diagram of the transmitter, channel, and receiver is shown in Figure 2 for the transmission of the i^{th} bit of user k in a coherent BPSK system; $\alpha_{k,l}^2$ is the Rayleigh fading variable for the k^{th} user in the l^{th} slot, and $I_{k,l}(t)$ indicates the combined interference (both multi-user and additive white Gaussian noise) that the k^{th} user sees on the l^{th} channel. Note that the system introduced here when the coding is simple repetition (or spreading) is identical to that put forward in [1, 7]. Besides the codeword across the L channels that a user occupies, it will also be assumed that the user is using some form of forward error correction in time. This error correction in time will normally be considered to be a complicated hard-decision decoded code, whereas the code across the channels will be considered a low complexity soft-decision decoded code. Thus, the system forms a simple concatenated coding system, identified only by the method in which the multi-user interference and diversity come into play. The optimization question being solved is simply a resource utilization question - is it worth the resource allocation cost to put extra redundancy in an inner soft-decision code that provides diversity gain and multi-user interference suppression?

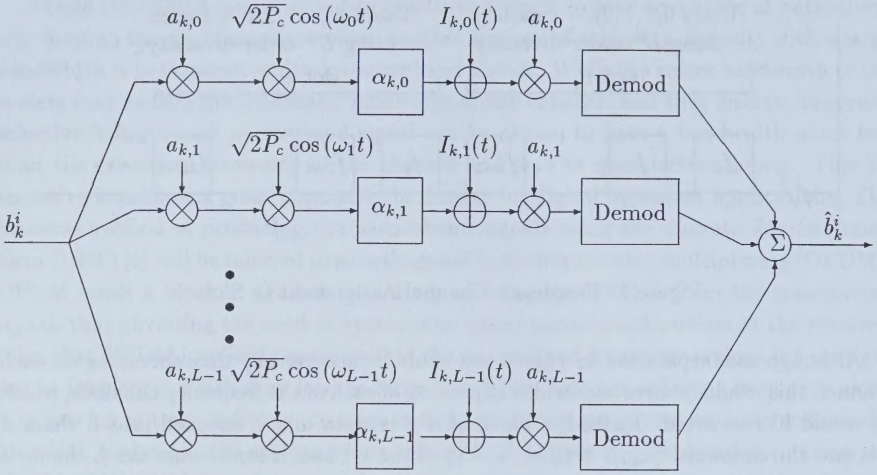

$a_{k,0}$
$\sqrt{2P_c}\cos(\omega_0 t)$
$\alpha_{k,0}$
$I_{k,0}(t)$
$a_{k,0}$
Demod
$a_{k,1}$
$\sqrt{2P_c}\cos(\omega_1 t)$
$\alpha_{k,1}$
$I_{k,1}(t)$
$a_{k,1}$
Demod
b_k^i
Σ
$\hat{b}_k^i$
$a_{k,L-1}$
$\sqrt{2P_c}\cos(\omega_{L-1} t)$
$\alpha_{k,L-1}$
$I_{k,L-1}(t)$
$a_{k,L-1}$
Demod

Figure 2: Effective Channel for User k

3 Probability of Error Calculations

The analytical work presented in this paper will deal with a coherent BPSK system using a correlation receiver on each frequency channel as shown in Figure 3. When the outer code is hard-decision decoded, the sufficient statistic for characterization of the inner system is simply the probability of error. In this section, the probability of error (or an approximation to it) is calculated for both a synchronous and an asynchronous system. In each case, the following assumptions are made:

1. Perfect channel estimates are available to the demodulator.

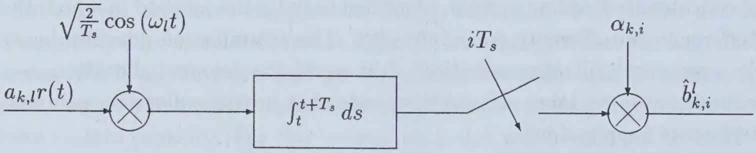

$\sqrt{\frac{2}{T_s}}\cos(\omega_l t)$
$a_{k,l}r(t)$
$\int_t^{t+T_s} ds$
iT_s
$\alpha_{k,i}$
$\hat{b}_{k,i}^l$

Figure 3: Correlator Receiver for Coherent BPSK, Perfect Fading Estimation

2. Each user is faded independently.

The second assumption is reasonable, for example, on the reverse link of a cellular system where the transmitters are in different locations. In this case, users would not be coordinated and thus asynchronous. However, in the succeeding section, the simpler synchronous case will be considered first with the extensions to the asynchronous case left to Section 3.2.

3.1 Probability of Error - Synchronous System

The following derivation is very similar to the derivation found in [4, pg. 723] for the single-user case. Without loss of generality, the reception of bit 0 of user 0 is considered. Under the assumptions of this paper, it is sufficient to consider the decision variable

$$\hat{b}_0^0 = \sqrt{\frac{E_s}{L}} \sum_{l=0}^{L-1} \alpha_{0,l}^2 + \sum_{l=0}^{L-1} \alpha_{0,l} \sqrt{\frac{E_s}{L}} \sum_{k=1}^{K-1} \alpha_{k,l} \cos\theta_{k,l} + \sum_{l=0}^{L-1} \alpha_{0,l}\eta_l \tag{1}$$

where E_s is the energy per user per symbol interval, $\theta_{k,l}$ is the phase difference between user k and user 0 on the l^{th} channel, and η_l is a zero-mean Gaussian random variable with variance $\frac{N_0}{2}$, which accounts for the noise in user 0's sample on the l^{th} channel. Let the received signal-to-background noise be denoted by $\Gamma = \frac{E_s}{N_0}\sigma_f^2$ where σ_f^2 is the second moment of the Rayleigh fading variables. It will be assumed that Γ is a fixed parameter of the channel (i.e. the parameter that labels a channel "good" or "bad", but which cannot be controlled by the system designer). Letting the "total fading" of the first user be given by $F = \sum_{l=0}^{L-1} \alpha_{0,l}^2$, $\hat{b}_0^0$ is Gaussian when conditioned on F. In other words, $p(\hat{b}_0^0 | F, b_0^0 = +1) \sim N(\sqrt{E_c}F, \sigma^2)$ where the variance of the noise and multi-user interference is given by $\sigma^2 = \frac{K-1}{2}\sigma_f^2 E_c F + \frac{N_0}{2}F$. Noting that F is chi-square with $2L$ degrees of freedom yields

$$P_e(K, L, \Gamma) = \int_0^\infty Q(\sqrt{2\bar{\gamma}_c f}) \frac{1}{(L-1)!} f^{L-1} e^{-f} df \tag{2}$$

where

$$\bar{\gamma}_c = \frac{\frac{E_s}{L}\sigma_f^2}{(K-1)\frac{E_s}{L}\sigma_f^2 + N_0} = \frac{\Gamma}{\Gamma(K-1) + L}$$

It is possible to evaluate the integral in (2) but the resulting expression is not well-suited to optimization over L and thus will not be used in succeeding sections.

3.2 Probability of Error - Asynchronous Systems

Let the delay of the k^{th} user relative to the desired user be given by $\tau_k T_s$ where τ_k is uniform in $(0,1)$. The exact probability of error can be calculated using methods similar to those in [2], but the expressions are complicated enough to be virtually useless for optimization. Invoking the Gaussian assumption on the multi-user interference given F, it is straightforward to show that the probability of error calculation is the same as in the previous section and thus

$$P_e(K,L,\Gamma) = \int_0^{\infty} Q(\sqrt{2\bar{\gamma}_c f})\frac{1}{(L-1)!}f^{L-1}e^{-f}df \tag{3}$$

with

$$\bar{\gamma}_c = \frac{\frac{E_s}{L}\sigma_f^2}{\frac{2}{3}(K-1)\frac{E_s}{L}\sigma_f^2 + N_0} = \frac{\Gamma}{\frac{2}{3}\Gamma(K-1)+L}$$

for the asynchronous case. It will be shown how it is possible to use the Gaussian case to consider asymptotic results that can be extended to the *actual* system and bound the performance of the best systems.

4 Throughput Optimization

4.1 Definition of Throughput

In this section, the throughput is formally defined for a system using an outer code that achieves the binary symmetric channel (BSC) capacity. The number of bits reliably transmitted per channel use for each user is given by $C(P_e(K,L,\Gamma))$, where $C(x) = 1-\mathcal{H}_2(x) = 1+x\log_2 x+(1-x)\log_2(1-x)$. Note that it has been assumed that successive bits for a given user are independent. This can be accomplished by perfect interleaving or by switching users pseudorandomly across frequency channels for the transmission of each codeword bit. The desired metric is "normallized throughput," measured in bits per channel per channel use, and expressed as

$$W(K,L,\Gamma) = \frac{qKC(K,L,\Gamma)}{N} = \frac{K}{L}C(P_e(K,L,\Gamma)) \tag{4}$$

The main goal of this work is to optimize W over K and L.

4.2 Optimization

4.2.1 Throughput at Low SIR Ratios - and the $K = 1$ result

To examine the case when $\bar{\gamma}_c \ll 1$, consider:

$$\begin{aligned} P_e(\bar{\gamma}_c) &= \int_0^\infty Q(\sqrt{2\bar{\gamma}_c x})\frac{x^{L-1}}{(L-1)!}e^{-x}dx \\ &= \int_0^\infty \lim_{P\to\infty}\sum_{p=0}^{P}\frac{Q^{(p)}(0)}{p!}(\sqrt{2\bar{\gamma}_c x})^p\frac{x^{L-1}}{(L-1)!}e^{-x}dx \end{aligned}$$

where $Q^{(p)}(0)$ is the p^{th} derivative of $Q(x)$ evaluated at $x = 0$. It is possible to show rigorously that the integral and limit can be exchanged if $\bar{\gamma}_c < 1$, and thus the probability of error for $\bar{\gamma}_c < 1$ is given by

$$\begin{aligned} P_e(\bar{\gamma}_c) &= \frac{1}{2} - \sqrt{\bar{\gamma}_c}\frac{(2L-1)!!}{2^L(L-1)!} + \frac{1}{2^L(L-1)!}\sum_{k=1}^{\infty}(-1)^{k+1}(\bar{\gamma}_c)^{k+\frac{1}{2}}\frac{(2k-1)!!(2k+2L-1)!!}{(2k+1)!} \\ &= \frac{1}{2} - \sqrt{\bar{\gamma}_c}\frac{(2L-1)!!}{2^L(L-1)!} + o(\bar{\gamma}_c) \qquad (5) \end{aligned}$$

Expanding the hard-decision capacity $C(x)$ about $x = \frac{1}{2}$, one obtains

$$C(x) = \frac{1}{\ln 2}\sum_{l=1}^{\infty}\frac{2^{2l}}{2l(2l-1)}(x-\frac{1}{2})^{2l}$$

Combining this result with (5), one obtains the throughput of the synchronous system as

$$W(K,L,\Gamma) = \frac{1}{2\ln 2}\left(\frac{\Gamma}{\Gamma L\left(\frac{(K-1)}{K}\right)+\frac{L^2}{K}}\right)\left(\prod_{l=1}^{L-1}(1+\frac{1}{2l})\right)^2 + o(\bar{\gamma}_c) \qquad (6)$$

If $K = 1$, the question to be answered is: for what Γ is it useful to take advantage of the diversity offered by $L > 1$ at the expense of added system resource usage? It is clear in the high-SNR limit where $P_e < C^{-1}(\frac{1}{2})$ that $L = 1$ is optimal because for $L > 2$, W is upper bounded by $\frac{1}{2}$. Using (6), it is possible to study the other extreme $\Gamma \ll 1$ to see if the answer changes. Letting $K = 1$ in (6), one obtains

$$W = \frac{\Gamma}{2\ln 2}\left(\frac{1}{L}\prod_{l=1}^{L-1}(1+\frac{1}{2l})\right)^2 + o(\Gamma)$$

The above function is strictly decreasing in L and thus is maximized at $L = 1$. Thus, at very low SNR, $L = 1$ is also optimal, leading one to conjecture that if the synchronous system employs repetition codes with one user per slot, $L = 1$ is optimal for all signal-to-noise ratios.

4.2.2 Asymptotics - Synchronous System

Consider now the case of a large number of channels and a large number of users. Observation of (6) leads to an interesting theorem:

Theorem 4.1 *Consider the binary coherent multicarrier system described in Section 2 with synchronous users. For any* $\Gamma > 0$,

$$\lim_{N\to\infty} \sup_{K,L} W(K,L,\Gamma) \geq \frac{2}{\pi \ln 2}$$

Proof: The rigorous proof is straightforward analysis on a system with $K = N^2, L = N$ and is omitted.

Since the above theorem shows how well one can do *regardless* of the signal-to-noise ratio, the next logical question is: what is the best one can do for any signal-to-noise ratio ? The following theorem answers this question.

Theorem 4.2 *Consider the binary coherent multicarrier system described in Section 2 with synchronous users. For any* $\Gamma > 0$,

$$\lim_{N\to\infty} \sup_{K,L} W(K,L,\Gamma) \leq \frac{4}{\pi}$$

Proof: The convexity of the $Q(\cdot)$ can be used to show this result. In actuality, plots suggest an upper bound of 1 achieved at $K = 1, L = 1, \Gamma = 0$, but this tighter upper bound has not been established rigorously yet.

4.2.3 Asymptotics - Asynchronous System

Theorem 4.3 *Consider the binary coherent multicarrier system described in Section 2 with asynchronous users and the conditional multi-user interference assumed to be Gaussian distributed. Then, for any* $\Gamma > 0$,

$$\lim_{N\to\infty} \sup_{K,L} W(K,L,\Gamma) \geq \frac{3}{\pi \ln 2}$$

Proof: The proof of this theorem is analogous to that of Theorem 4.1 and also omitted.

The above theorem's reliance on the Gaussian multiple access interference assumption is a glaring weakness. This is overcome by the following theorem, the proof of which relies heavily on the theorem above.

Theorem 4.4 *Consider the binary coherent multicarrier system described in Section 2 with asynchronous users. For any* $\Gamma > 0$*,*

$$\lim_{N\to\infty} \sup_{K,L} W(K,L,\Gamma) \geq \frac{3}{\pi \ln 2}$$

Proof: The idea of the proof is quite simple. Using a system with $K = N^2, L = N$, one lets the variance of the multi-user be a large fixed constant. Next one applies a strong uniform central limit theorem to reduce this theorem to that of the previous therem. Finally, one lets the variance of the multi-user interference be large enough to establish the result within any $\epsilon > 0$.

5 Conclusions and Continuing Work

In this paper, a model has been formulated that considers concatenated coding systems operating over independent fading channels. The key question of whether to put the redundancy in a simple soft-decision decoded inner code or into a sophisticated hard-decision decoded outer code to optimize system throughput is introduced. Motivation for this type of work is found directly in the current literature on OFDM systems if the infinite interleaving assumption is made.

The key results in this paper have considered the aforementioned optimization for a coherent system employing repetition inner codes and capacity-achieving outer codes. For a synchronous system, the exact probability of error has been calculated while for an asynchronous system, a similar error calculation requires a Gaussian assumption on the conditional multi-user interference. In each case, these results allow one to obtain tight bounds on the throughput for the *actual* system, essentially solving the optimization problem over the number of users and slot size.

The continuing work on this project for a coherent system consists of optimization of the throughput over the diversity of each user when the number of users in the *system* is fixed. Parallel work for the equally important noncoherent case is also being considered. Finally, the observant reader may note that the multicarrier system bears more than slight similarity to both frequency-hopped and direct-sequence spread-spectrum, and the authors are exploring whether the multicarrier formulation presented here can be used to unite the two concepts analytically.

References

[1] G. Fettweis, A. Bahai, and K. Anvari, "On Multi-Carrier Code Division Multiple Access (MC-CDMA) Modem Design," *Proceedings of the 1994 Vehicular Technology Conference,* pp. 1670-1674, June 1994.

[2] E. Geraniotis and M. Pursley, "Performance of Coherent Direct-Sequence Spread-Spectrum Communications Over Specular Multipath Fading Channels," *IEEE Transactions on Communications,* Vol. COM-33, pp. 502-508, June 1985.

[3] I. Gradshteyn and I. Rhyzhik, *Table of Integrals, Series, and Products,* Academic Press, New York, 1980.

[4] J. Proakis, *Digital Communications,* McGraw-Hill, New York, 1989.

[5] D. Rowitch and L. Milstein, "Coded Multicarrier Code Division Multiple Access," *Transactions of the 1995 International Symposium on Information Theory,* pg. 23, September 1995.

[6] J. Salz and S. Weinstein, "Data Transmission by Frequency-Division Multiplexing Using the Discrete Fourier Transform," *IEEE Transactions on Communications Technology,* vol. COM-19, pp. 628-634, October 1971.

[7] N. Yee and J. Linnartz, "Controlled Equalization of Multi-Carrier CDMA in an Indoor Rician Fading Channel," *Proceedings of the 1994 Vehicular Technology Conference,* pp. 1665-1669, June 1994.

Data Communications in a Code-division Multiple-access System *

Daniel L. Noneaker and Michael B. Pursley

Clemson University
102 Riggs Hall, Box 340915
Clemson, SC 29634-0915 USA

Abstract

Results are presented for the probability of packet error in a mobile code-division multiple-access (CDMA) system with binary convolutional coding and block interleaving. Packet-data communications is considered both for reverse links with closed-loop power control and reverse links without closed-loop power control. The effectiveness of coding and interleaving is compared for systems with different chip rates. The effectiveness of different strategies for adapting the coding and interleaving to channel conditions is also considered.

I. Introduction

Direct-sequence (DS) spread spectrum is used in mobile cellular code-division multiple-access (CDMA) networks which are currently under development [1], and it will soon appear in several cellular systems and personal communication systems [2]. In addition, it has been implemented for peer-to-peer mobile communications in distributed packet radio networks with no base station [3]. The design of distibuted CDMA networks has focused on packet data services since such networks were first proposed. In contrast, the development of emerging cellular CDMA systems has focused primarily on the support of voice communications. However, the increasing demand for packet data services points to the need for cellular systems which support both voice and data traffic efficiently.

CDMA systems are subject to near-far interference [4], and its effects can be mitigated by the use of power control [5]. If a transmitter receives timely feedback from the receiver, closed-loop power control can be employed to rapidly adapt the power of the transmitter to the channel response or the interference environment [5]. The reverse (mobile to base station) link in a cellular CDMA system is particularly susceptible to near-far interference, and it is generally conceded to be limiting link in the system. If full-duplex voice communication occurs between a base station and a mobile, the forward (base station to mobile) link serves as a natural feedback channel for base-station control of the power of the mobile transmission. Similarly if a connection is established for continuous transmission of data on the reverse link, a forward-link control channel can be used in part to support power-control commands.

---

*This research was supported by the Holcombe Endowment at Clemson University and by the Army Research Office through the Focused Research Initiative grant DAAH04-95-1-0247.

Even if the connection consists of asynchronous bursts of packets, closed-loop power control may be effective during at least a portion of each burst.

However, in many circumstances rapid power control is not feasible. If data packets are transmitted using a random-access protocol, there is no efficient means of providing timely feedback during the transmission of a packet. In a distributed CDMA network, attempts at rapid closed-loop power control are likely to lead to network instability and poor performance, even if timely feedback *is* available. In both of these situations, the mobile must determine *a priori* the appropriate power level to use for the duration of the data packet. If a constant power level is used to transmit a packet, forward error-correction coding with the interleaving of code symbols is the primary means of compensating for rapid fading of the channel.

In this paper, we examine the effects of forward error-correction coding and interleaving on the performance of data communications in a CDMA system. Two types of data transmission are considered. One consists of continuous transmission of data packets on a link in a cellular in a cellular CDMA system. The mobile and base station engage in an extended connection, and the base station maintains a feedback channel to the mobile which is used for closed-loop power control for the duration of the connection. The other type of data transmission employs a random-access protocol, and there is no closed-loop power control. The results for the second type of transmission are applicable to both cellular and decentralized CDMA networks. The effectiveness of coding and interleaving is examined for different code rates, for systems with different chip rates, and for both types of data transmission. Strategies for adapting the coding and interleaving to channel conditions are discussed.

II. Doubly Selective Fading Channels

The mobile communications channel often consists of a few strong paths (perhaps only one) and a number of weaker paths [6]. For a DS system with a low chip rate, the delays for multiple multipath components fall within one chip interval. The composite signal formed by such components may be modeled as a single diffuse fading signal which has a Rician fading amplitude with a strong Rayleigh component and a relatively weak specular component. However, if the chip interval is much smaller than the smallest difference among the delays for the various propagation paths, a correlation receiver can lock onto a single multipath component. Thus, the channel may appear to the correlation receiver to produce a single component which exhibits Rician fading with a strong specular component and a relatively weak Rayleigh component or no Rayleigh component at all.

We model the channel in a manner that reflects the phenomenon described above and allows for tractable analysis of the receiver. The channel represents the collection of propagation paths from transmitter to the receiver. It consists of several clusters of paths, with each cluster having a single specular path plus a continuum of paths that collectively exhibit Rayleigh fading and for which the path delays which are centered at the delay of the specular path. The delay spectrum of the channel is illustrated in Figure 1. The channel is a special case of the Gaussian wide-sense-stationary uncorrelated-scattering (WSSUS) channel [7]. The rate of variation of the channel response is given by the normalized *Doppler spread* D_T of the channel, and it is given by $D_T = B_d T$ where B_d is the Fourier transform of the time-correlation function of the channel and T is the duration of a channel symbol. For a given delay spectrum and a given depth of fading, the Doppler spread is inversely proportional to the average duration of a fade [8]. The channel model is described in more detail in [9].

III. Packet Format and Receiver Architecture

Each packet in a random-access transmission consists of an acquisition header of fixed length followed by encoded data consisting of one or more blocks of interleaved code symbols. The blocks contain all packet-header information in addition to the data payload. In the numerical results given in this paper, the overhead is modeled as equivalent to the cost of transmitting 576 code symbols. In contrast, the continuous transmission consists of a sequence of fixed-length packets, each of which contains the sequence number and the data payload. No acquisition header is necessary for the packets with continuous transmission, and the overhead is modeled as having negligible cost.

The system employs binary convolutional encoding and block interleaving. The output of the encoder for a packet is read by columns into blocks of dimensions m columns by n rows. The contents are read out by rows, so that the *interleaving depth* is n. For each code that is considered, the row dimension of the interleaving block is chosen to match the maximum span among the minimum-distance error events of the code. Thus for a given block size, the greatest possible time diversity is provided as protection against those error events. Differential phase-shift keyed modulation follows the interleaver. The receiver is a rake receiver with post-detection equal-gain square-law diversity combining. It is assumed that the ith tap of the receiver is synchronized to the signal component arriving with a delay corresponding the center of the ith cluster of the channel. The receiver employs noncoherent demodulation, and each sample from the output of the square-law detector results from the signal received over two consecutive data pulse intervals [9]. The receiver employs deinterleaving followed by Viterbi decoding with the square-law metric which is produced at the output of the rake receiver.

If closed-loop power control is used, the base-station receiver estimates the signal-to-interference ratio at the output of the rake receiver, compares it with a target value, and generates a power-control command based on the comparison. The interval between power-control commands is 30 T, and there is a delay of 60 T between the generation of an estimate at the base station and the execution of the corresponding power-control command at the mobile. The mobile transmitter either raises or lowers its power level by 1 dB in response to each power-control command.

III. Probability of Packet Error

In order to characterize completely the performance of data communications in a CDMA system it is necessary to determine the throughput of each link, the aggregate throughput of the system, and the delay characteristics of each link. If voice communication is supported, the impact of data traffic on the delivery of voice traffic must also be evaluated. For a given propagation environment and traffic profile, these performance measures depend on the packet format (including coding and interleaving), the channel-access protocol, the power-control technique, and the acquisition, demodulation, and and decoding techniques used in the base-station receiver.

However, much insight into the performance of the system can be obtained by considering the probability of packet error as a function of the average signal-to-interference ratio. This performance measure is determined by the characteristics of the channel, the format of the packet, the chip rate of the system, the power-control technique, and the acquisition, demodulation, and and decoding techniques. In practice the system would also employ an outer error-detection code, so that most packet errors result in a retransmission attempt. Our measure of the packet error probability is a close approximation to the decoder-failure rate in such a system, and thus it determines the throughput and delay which result with a particular channel-access protocol under a given traffic load. The average probability of packet error is the performance criterion that is used in this paper.

Packet errors in a continuous data transmission are due to decoder errors. For random-access transmission, some packet errors are due to failure to acquire the packet. However, errors due to missed acquisition are not considered in this paper. In the examples given in this paper, the entire packet content is contained in one interleaving block.

The performance of the system is assumed to be limited by multiple-access interference, and the composite interference at the output of each correlator is modeled as additive white Gaussian noise with single-sided power spectral density I_0. For the examples given in this paper, I_0 is constant throughout the duration of an interleaving block. The average signal-to-interference ratio at the output of the rake receiver is denoted SIR, where the signal energy is the received energy per bit of information averaged over the Rician fading of the channel. Conditioned on the information sequence, the square-law statistic produced in each code-symbol interval is a function of a vector of jointly Gaussian complex-valued random variables determined by the fading process, the chip rate, the code-symbol rate, the number of taps in the rake receiver, and the transmitter power level [9].

The average probability of error at the input to the decoder can be expressed in closed form [9] if no closed-loop power control is used, and it can be expressed in closed form for some models of closed-loop power control [10]. In contrast, no closed-form expression can be obtained for the probability of error at the *output* of the decoder. In some instances, transfer-function bounds can be used to determine the performance of coding and interleaving [11]. However, the bounds fail to converge for many circumstances of interest in mobile communications. Therefore in this paper, simulations are employed to examine the effect of fading on the performance at the output of the decoder for CDMA systems with convolutional coding and interleaving.

IV. The Effect of the Chip Rate on Performance

The chip rate of a CDMA system can significantly affect the probability of error at the input to the decoder, both for systems with rapid, closed-loop power control [10] and for systems without closed-loop power control [9]. In this section, we determine the effect of the chip rate on the packet error probability at the decoder output for a Rician-fading channel with the delay profile illustrated in Figure 1. The intent is *not* to compare the absolute level of performance of systems with different chip rates, and SIR as defined here is not appropriate for that purpose [12]. Instead, we consider the way in which the chip rate affects the *sensitivity* of the system to the Doppler spread, the packet size, and the number of taps in the rake receiver.

The channel has three clusters of paths, and each cluster has a total delay spread of $0.05\,T$. The first cluster has a specular-to-diffuse energy ratio of -3 dB. The other two clusters have no specular component, but each has a diffuse component with an average power that is 6 dB less than the average power in the diffuse component of the first cluster. The parameters are chosen to model conditions which are typical of urban multipath channels.

The normalized chip rate of the system is $R = T/T_c$, where T_c is the chip duration. The value $R = 50$ is taken as a representative low chip rate, and $R = 400$ is used as an example of a high chip rate. The rate one-half convolutional code that is considered has a constraint length of seven and a minimum free distance of ten [13]. In each example, the performance measure is the value of SIR that is required to achieve an average probability of packet error of 10^{-2}.

The effect of the chip rate on the performance of a link without closed-loop power control is illustrated in Figure 2. The required SIR is shown as a function of the Doppler spread for the low chip rate and a three-tap rake receiver and for the high chip rate and a correlation receiver. Blocks interleavers of dimensions 18 by 32 and 18 by 128 are considered. Though not shown here, the low-chip-rate system is susceptible to persistent deep fades if a correlation receiver is used, and the coding and interleaving do not provide adequate time diversity to overcome the fading. In contrast, the use of a high chip rate provides effective multipath discrimination so that the output of the correlator is rarely faded. Thus, for almost all packets, the signal-to-interference ratio at the input to the decoder of the high-chip-rate system remains high enough to obtain coding gain, and the system is relatively insensitive to the Doppler spread. Consequently, the additional interleaving depth provided by a larger packet size has little effect on the performance. For the high chip rate, the difference in performance for the two packet sizes is due almost entirely to the difference in efficiency resulting from the packet overhead.

For this example, the rake receiver provides little performance improvement in the high-chip-rate system. In contrast, the susceptibility to fading of the low-chip-rate system is reduced if a rake receiver is employed in place of a correlation receiver. The additional taps provide multipath diversity, and fading occurs less frequently at the output of the rake combiner than at the output of the correlator. However, rake reception does not completely eliminate deep fades with the low chip rate, and the system is still susceptible to persistent fades. Thus the performance with an 18 by 32 block size is 4.7 dB worse if the Doppler spread is 0.0002 than if the Doppler spread is 0.008. An increase in the block size and the corresponding interleaving depth significantly increases the time diversity provided by the code at intermediate values of the Doppler spread, and it consequently yields a significant improvement in performance for the low-chip-rate system.

The effect of the chip rate on the performance of a continuous data transmission with closed-loop power control is shown in Figure 3 for an 18 by 32 block interleaver. At low Doppler spreads, closed-loop power control is effective in maintaining the signal-to-interference ratio at the combiner output at a nearly constant value, and it drastically reduces the susceptibility of the system to fading. In particular, closed-loop power control significantly improves the performance of the low-chip-rate system in persistent fading. Neither power control nor coding and interleaving are entirely effective at intermediate values of Doppler spread, and worst-case performance with the low chip rate occurs for this range of values. An increase in the block size reduces the minimum Doppler spread for which coding and interleaving is effective, and a sufficiently large increase in the block size ensures that power control and coding together provide adequate protection against fading over the full range of Doppler spreads. For example, the performance of the low-chip-rate system improves significantly if an 18 by 128 block is used instead of an 18 by 32 block, and the sensitivity to the Doppler spread is reduced. In contrast, the performance of the high-chip-rate system varies by less than 1 dB over the range of Doppler spreads up to 0.01 even if an 18 by 32 interleaver is employed. The use of a larger interleaving block yields little performance improvement with the high chip rate, since the system is not susceptible to deep fading and packet overhead is not a significant factor for continuous transmission.

V. The Effect of the Code Rate on Performance

In addition to the influence of the chip rate, the channel characteristics, and the number of taps in the rake receiver, the information rate that can be achieved on a link depends on the depth of interleaving, the distance properties of the code, and the signal-to-interference ratio at the receiver. In this section, we take the viewpoint that information rate, transmitted energy per bit of information, and block duration (and consequent decoding delay) are the fundamental parameters of the packet format. Any or all of the parameters may be changed in response to changes in the quality of the channel. We consider different combinations of the code rate and the channel-symbol rate that are consistent with a given set of those fundamental parameters in order to determine the effect of code selection on the packet error probability. (The appropriate choice of interleaving depth is determined by the distance properties of the code, the code-symbol rate, and the block duration.) The results are used to examine methods for adapting the information rate in a system which is constrained to use a fixed channel-symbol rate. As demonstrated in the previous section, the low-chip-rate system is more sensitive than the high-chip-rate system to the variation of system parameters, so we focus on the low chip rate in this section.

The convolutional codes that are considered are the rate-$1/n$, constraint-length seven codes of largest minimum free distance [13], and comparisons of the rate-one-quarter code with the rate-one-half code are discussed in detail. It is shown in [14] that, by decreasing the code rate, nearly optimal performance can be approached with coherent demodulation in an additive, white Gaussian noise (AWGN) channel at the expense of decoder complexity which increases with decreasing code rate. However, we compare codes of the same constraint length, since this provides a fair comparison on the basis of decoder complexity. It is easily shown that the minimum free distance of a binary, rate-k/n convolutional code is bounded above by

$$d_{free} \leq \left\lfloor \frac{2K+2}{3} n \right\rfloor$$

where K is the constraint length [15] of the code. The bound is fairly tight for the best rate-$1/n$ codes of practical interest in mobile communications. Thus the minimum free distances of the optimal codes of a given constraint length are approximately proportional to the inverse of the code rate so that, if the energy per bit of *information* is fixed, each code yields approximately the same minimum Euclidean distance. A lower-rate code has a weight spectrum which is more sparse than a higher-rate code, but it is obtained at the expense of greater noncoherent combining loss in a system with noncoherent demodulation.

In contrast, the increase in the minimum Hamming distance that results from lowering the code rate *does* increase in the time diversity provided by coding, if the channel symbols are subject to independent fading [16]. However for a given probability of error and a given interleaving depth, the effective order of time diversity provided by any code approaches one as the Doppler spread of the channel approaches zero. Thus a lower-rate code may not provide much more diversity than a higher-rate code if the fading is persistent.

Consider the performance of random-access transmissions for three packet formats that employ the same information rate and a block duration of 576 T. Packet format A uses the rate-one-half code, a channel-symbol rate of T^{-1}, and a block size of 18 by 32. In packet format B, the rate-one-fourth code is employed together with a 36 by 32 interleaver and a channel-symbol rate of $(T/2)^{-1}$. Packet format C uses a rate-one-fourth code that is formed by the concatenation of the rate-one-half encoder with a dual repetition encoder.

The interleaver dimensions and channel-symbol rate for format C are the same as for format B. The performance for formats A and B are shown in Figure 4.

If the Doppler spread is small, the greater noncoherent combining loss of the rate-one-fourth code in comparison to the rate-one-half code offsets the effects of the superior weight spectrum of the lower-rate code, and packet format A yields better performance than packet format B. Even if the Doppler spread is large, the coherence time of the channel is greater than the interleaving interval. In addition, a minimum-weight error event of the low-rate code results from channel symbols spanning approximately the same time interval as the channel symbols that result in a minimum-weight error event of the high-rate code. Thus the diversity obtained with the lower code rate is not much greater than the diversity obtained with the higher code rate, and format A yields slightly better performance than format B even for the largest Doppler spread of interest. However, the difference in the performance of the two formats is small over the range of Doppler spreads. The performance with format C is uniformly poorer than that with format B, but the two differ by less than 0.3 dB for any value of the Doppler spread.

A similar comparison is illustrated in Figure 4 for a block of duration 2,304 T. Three packet formats, D, E, and F, are considered which differ from formats A, B, and C, resp., only in that the dimensions of the interleavers for D, E, and F are 18 by 128, 36 by 128, and 36 by 128, resp. The greater block duration allows each interleaver to more closely approximate the condition of independent fading for large Doppler spreads so that the difference in the time diversity obtained with the two codes is greater than the difference for the smaller block size. Thus packet format E yields slightly better performance than packet format D for the largest values of Doppler spread. The penalty for using the concatenated code in place of the optimal rate-one-fourth code is small for all Doppler spreads.

The desire for a simple reverse-link receiver design may dictate that the system operate with a predetermined channel waveform and channel-symbol rate. In that case the information rate can be changed only by varying the code rate, and several codes may be required in order to provide adequate flexibility for the adaptation protocol. If the optimal code of each rate is used, the Viterbi decoder must be sufficiently general to allow the decoding of any of the codes. If, instead, a single high-rate encoder is used and lower-rate codes are generated by repetition of the high-rate code, then a more cost-effective implementation of the decoder may be possible. As seen in the results above, very little performance is sacrificed with this approach.

VI. Conclusions

We have examined the performance of convolutional coding and interleaving in a CDMA mobile packet-data communication system. The results show that the performance of a CDMA system with a low chip rate is much more sensitive to the interleaving depth and the Doppler spread than is the performance of a system with a high chip rate. In particular, the low-chip-rate system is vulnerable to persistent fading in situations for which closed-loop power control is not feasible. It is also shown that the performance does not depend critically on the code rate *per se*, but it depends, rather, on the information rate, the energy per bit of information, and the duration of the interleaving block. Variable-rate coding can be implemented by concatenating a fixed encoder with a variable-rate repetition encoder, and this technique results in little performance loss compared with the use of the optimal code of each rate.

Acknowledgement

The authors wish to thank Dr. Colin Frank of Motorola Corporate Research for providing a portion of the software employed to generate the numerical results.

References

[1] IS-95, "Mobile station – base station compatability standard for dual-mode wideband spread spectrum cellular system," *Telecommunications Industry Association/ Electronics Industry Assocation Interim Standard*, July 1993.

[2] C. Cook, "Development of air interface standards for PCS," *IEEE Personal Communications*, pp. 30–34, Fourth Qtr. 1994.

[3] J. Jubin and D. Tornow, "The DARPA packet radio network protocols," *Proc. IEEE*, vol. 75, pp. 21–32, Jan. 1987.

[4] M. B. Pursley, "The role of spread spectrum in packet radio networks," *Proc. IEEE*, vol. 75, pp. 116–134, Jan. 1987.

[5] A. J. Viterbi, A. M. Viterbi, and E. Zehavi, "Performance of power-controlled wideband terrestrial digital communication," *IEEE Trans. Commun.*, vol. COM-41, pp. 559–569, Apr. 1993.

[6] D. L. Schilling and L. B. Milstein, "Broadband CDMA for indoor and outdoor personal communication," *Proc. of the Fourth Internat. Symp. on Personal, Indoor, and Mobile Commun.*, Sept. 1993.

[7] P. A. Bello, "Characterization of randomly time variant linear channels," *IEEE Trans. Commun. Syst.*, vol. CS-11, pp. 360–393, Dec. 1963.

[8] W. C. Jakes, ed., *Microwave Mobile Communications.* New York: IEEE Press, 1993.

[9] D. L. Noneaker and M. B. Pursley, "On the chip rate of CDMA systems with doubly selective fading and rake reception," *IEEE Journal on Selected Areas in Communications*, vol. 12, pp. 853–861, June 1994.

[10] J. Gass, D. L. Noneaker, and M. B. Pursley, "On the spectral efficiency of a power-controlled CDMA mobile personal communication system," to appear in the *IEEE Journal on Selected Areas in Communications, Special Issue on Wireless Local Communications.*

[11] D. L. Noneaker and C. Frank, "The effect of finite interleaving depth on the performance of convolutional codes in Rician-fading channels," *Proc. of the 1994 IEEE International Symposium on Information Theory*, p. 29, June 1994.

[12] J. H. Gass, D. L. Noneaker, and M. B. Pursley, "The performance of voice and data communications in a mobile cellular CDMA system," *Proc. of the 1995 IEEE International Symposium on Information Theory*, p. 24, Sept. 1995.

[13] S. Lin and J. D. J. Costello, *Error Control Coding: Fundamentals and Applications.* Englewood Cliffs, NJ: Prentice-Hall, 1983.

[14] A. J. Viterbi, "Very low rate convolutional codes for maximum theoretical performance of spread-spectrum multiple-access channels," *IEEE Journal on Selected Areas in Communications*, vol. 8, pp. 641–649, May 1990.

[15] A. J. Viterbi and J. K. Omura, *Principles of Digital Communications and Coding.* New York: McGraw-Hill, 1979.

[16] D. L. Goeckel and W. E. Stark, "Performance of coded direct-sequence systems in fading channels with rake reception," *Proc. of the 1994 IEEE Military Communications Conf.*, pp. 791–795, Oct. 1994.

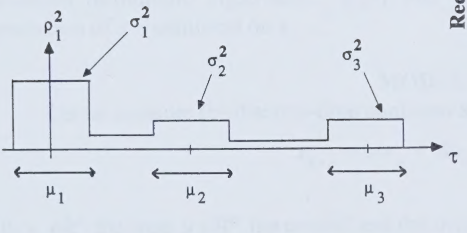


Figure 1. Delay profile of channel.

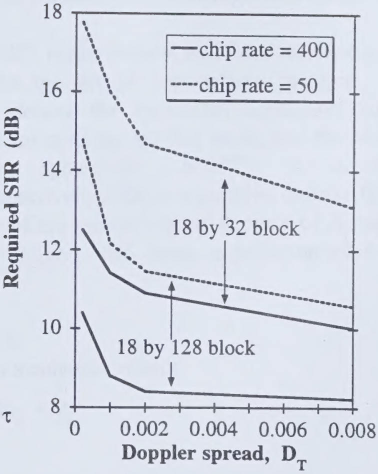


Figure 2. Performance without closed-loop power control.

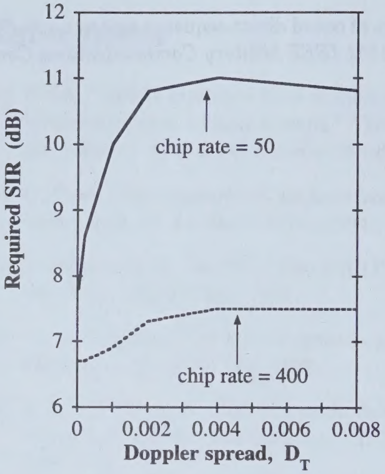


Figure 3. Performance with closed-loop power control.

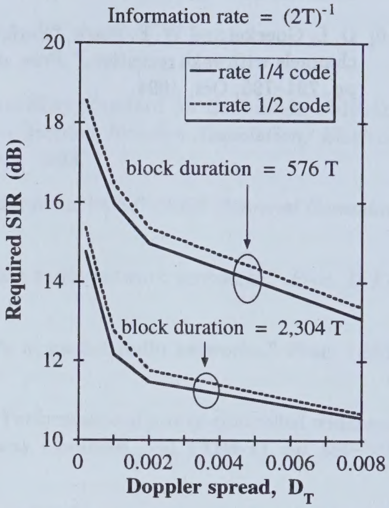


Figure 4. Comparison of four block formats without closed-loop power control.

ON SOME LMI PROBLEMS ARISING IN NONLINEAR STOCHASTIC CONTROL

Edwin Engin Yaz
Department of Electrical Engineering, University of Arkansas
Fayetteville, AR 72701. ph.: (501) 575-6580, fax: (501) 575-7967
e-mail: ey1@engr.uark.edu

Abstract. This work involves formulation of some problems arising in the analysis and control of a class of discrete-time nonlinear stochastic systems in terms of linear matrix inequalities. This allows one to utilize a wide variety of numerical techniques available for tackling both the feasibility problem and the actual numerical solution in an efficient manner.

INTRODUCTION

In this work, we consider a class of discrete-time nonlinear stochastic system models where the nonlinear functions are described by statistical means [1]-[3]. It is shown that discrete state space models with multiplicative white noises are subclasses of these models. Typical analysis and control problems will be considered and their solutions will be formulated in terms of linear matrix inequalities (LMIs). We will start out with analysis problems like stability, bounds on the state vector for ℓ_2 type inputs, bounds on ℓ_2 gain of the system, and bounds on output energy for given initial states, all of which are re-interpreted for this class of stochastic models. Then, stochastic versions of control problems like state feedback stabilizability and ℓ_2 gain minimization will be investigated. Some extentions of these results will also be indicated. These formulations will be hoped to lead to efficient numerical solutions of the corresponding problems.

The following notation will be used. For $x \in \mathbb{R}^n$, n-dimensional real Euclidean space, $\|x\|$ will be the 2-norm of the vector. Z^+ will be the set of nonnegative integers. For $A \in \mathbb{R}^{nxm}$, A^T, $A^{[2]}$, $st(A)$, $A^\dagger$ will respectively denote the transpose, Kronecker square (Kronecker product of A with itself), stack or vector operator applied on A, and the Moore-Penrose pseudo-inverse of A. For symmetric $A \in \mathbb{R}^{nxn}$ $A > (A \geq 0)$, A^{-1}, $A^{1/2}$, $A^{-1/2}$, $tr[A]$, $\lambda_{\max}(A)$ ($\lambda_{\min}(A)$) will respectively indicate a positive definite (semi-definite) A, its inverse, square root of A, inverse of the square root of A, trace of A, and its maximum (minimum) eigenvalue. $E\{x\}$ and $E\{x/y\}$ will mean expectation of x and expectation of x conditional on y.

MODEL

Let us consider the discrete-time nonlinear stochastic system

$$x_{k+1} = Ax_k + Bu_k + f_k \tag{1}$$

with $x_k \in \mathbb{R}^n$ the state, $u_k \in \mathbb{R}^m$ the control and the measurement equation

$$y_k = Cx_k + Du_k + g_k \tag{2}$$

where $f_k = f(x_k,\ k)$: $\mathbb{R}^n \times Z^+ \rightarrow \mathbb{R}^n$ and $g_k = g(x_k, k)$: $\mathbb{R}^n \times Z^+ \rightarrow \mathbb{R}^p$ are described by their statistical characteristics as follows:

$$E\left\{\begin{bmatrix} f_k \\ g_k \end{bmatrix} / x_k\right\} = 0, \; E\left\{\begin{bmatrix} f_k \\ g_k \end{bmatrix}\begin{bmatrix} f_j^T & g_j^T \end{bmatrix} / x_k\right\} = 0, \; k \neq j \tag{3}$$

$$E\left\{\begin{bmatrix} f_k \\ g_k \end{bmatrix} [f_k^T \; g_k^T] / x_k\right\} = \sum_{i=1}^{q} \begin{bmatrix} \Phi_{11}^i & \Phi_{12}^i \\ (\Phi_{12}^i)^T & \Phi_{22}^i \end{bmatrix} x_k^T \Gamma^i x_k \tag{4}$$

As can be checked easily, e.g. linear models containing multiplicative noise elements like $d^T x_k B v_k$ where v_k is a zero mean white noise vector $d \in \mathbb{R}^n$, $B \in \mathbb{R}^{nxn_v}$ belong to the class of models described above. Also included are nonlinear stochastic models containing e.g. $\|x_k\| B v_k$, $sgn[\phi(x_k)] B v_k$ where the signum function is applied to the nonlinear function ϕ: $\mathbb{R}^n \to \mathbb{R}$, and/or $|b^T x_k| B v_k$. Quadratic optimal control of such models was derived in [1] and [3] and the linear minimum variance estimator was obtained in [2]. A possible generalization of this model will be indicated later on.

STABILITY

The system (1) with $u_k \equiv 0$, $k \geq 0$ is said to be mean square (m.s.) asymptotically stable if $\lim_{k\to\infty} E\{\|x_k\|^2\} = 0$ for every $x_0 \in \mathbb{R}^n$ [4]. It is said to be m.s. exponentially stable if there exist $\alpha \in (0,1)$, $\beta > 0$ such that $E\{\|x_k\|^2\} \leq \beta\alpha^k$, $k \geq 0$ for every $x_o \in \mathbb{R}^n$. System (1) is almost surely (a.s.) exponentially stable if for any $x_o \in \mathbb{R}^n$, $\|x_k\| \leq \beta\alpha^k$ with probability one, for $k \geq 0$ and some $\alpha \in (0,1)$, $\beta > 0$. We have the following stability results:

Theorem 1. The following statements are equivalent

$$i) \quad \rho(\psi) < 1 \tag{5}$$

where

$$\psi = A^{T[2]} + \Sigma_{i=1}^{q} st(\Gamma^i)\, st^T(\Phi_{11}^i) \tag{6}$$

ii) there exists $P > 0$ such that

$$P - A^T P A - \Sigma_{i=1}^{q} \Gamma^i tr[\Phi_{11}^i P] > 0 \tag{7}$$

iii) there exists $Q > 0$ such that

$$Q - AQA^T - \Sigma_{i=1}^{q} \Phi_{11}^i \; tr[Q\Gamma^i] > 0 \tag{8}$$

iv) the system (1) is m.s. asymptotically stable

v) the system (1) is m.s. exponentially stable

Proof. $i \Leftrightarrow ii$: Writing ii) in the equivalent form

$$P = A^T P A + \Sigma_{i=1}^{q} \Gamma^i tr[\Phi_{11}^{i} P] + R \tag{9}$$

for some $R > 0$ and considering this as the steady state solution to

$$P_{k+1} = A^T P_k A + \Sigma_{i=1}^{q} \Gamma^i \, tr[\Phi_{11}^{i} P_k] + R \tag{10}$$

where "k" is the discrete-time index, by the use of Kronecker products and stack operator, it can be shown that the difference equation converges to $P > 0$ if and only if i) holds.
$ii) \Rightarrow v)$ Using $P > 0$ in $V_k = x_k^T P x_k$ in a stochastic Lyapunov (super-Martingale) analysis gives v) from ii).
$ii) \Leftrightarrow iii)$ Using the equivalent form (9) of (7), applying the stack operator to both sides and using the fact that the eigenvalues of a matrix and its transpose are the same, the equivalence is shown.
$v) \Rightarrow iv)$ Trivially follows from the definitions.
$iv) \Rightarrow iii)$ Can be done similarly to that given in [5].
The above conditions are sufficient for asymptotic stability in the mean $(\ell im_{k \to \infty} E\{x_k\} = 0)$ and a.s. exponential stability [4].

BOUNDS ON STATE FOR UNIT ENERGY INPUTS

Consider the system (1) with $x_0 = 0$ where u_k is an ℓ_2 type exogeneous disturbance input with unit energy $\Sigma_{k=0}^{\infty} \|u_k\|^2 \leq 1$. Based on this, the following bound on the second moment of the system state will be found:
Theorem 2. Let there be $P > 0$ such that

$$\begin{vmatrix} A^T P A + \Sigma_{i=1}^{q} \Gamma^i tr[\Phi_{ii}^{i} P] & A^T P B \\ B^T P A & B^T P B \end{vmatrix} - \begin{bmatrix} P & 0 \\ 0 & I \end{bmatrix} \leq 0 \tag{11}$$

then the state of system (1) with $x_o = 0$ and the unit energy input u_k satisfies $E\left\{x_N^T P x_N\right\} \leq 1$ for all $N \geq 0$.
Proof. The proof is done in two steps. First, it is shown that (11) implies for $V_k = x_k^T P x_k$ that

$$E\left\{V_{k+1}/x_k\right\} - V_k \leq \|u_k\|^2 \tag{12}$$

or

$$E\{V_{k+1} - V_k\} \leq \|u_k\|^2 \tag{13}$$

which yields

$$\Sigma_{k=o}^{N-1} E\left\{V_{k+1} - V_k\right\} \leq \Sigma_{k=0}^{N-1} u_k^T u_k \tag{14}$$

So, for any $N \geq 0$, we obtain the result.

If the exogeneous input u_k is of stochastic nature with

$$\Sigma_{k=0}^{\infty} E\left\{\|u_k\|^2\right\} \leq 1 \tag{15}$$

then we have the following result:

Theorem 3. The results of theorem 2 hold under the same conditions for stochastic inputs u_k satisfying (15).

Proof. The proof of this result is almost the same except that (ii) implies that

$$E\left\{V_{k+1}/x_k, u_k\right\} - V_k \leq \|u_k\|^2 \tag{16}$$

which yields

$$\Sigma_{k=0}^{N-1} E\left\{V_{k+1} - V_k\right\} \leq \Sigma_{k=0}^{N-1} E\{\|u_k\|^2\} \tag{17}$$

to replace (14) that gives the result.

Note that since we have

$$\lambda_{\min}(P) E\{\|x_k\|^2\} \leq E\{x_k^T P x_k\} \leq 1 \tag{18}$$

it is possible to find upper bounds on $E\{\|x_k\|^2\} = tr[X_k]$ where $X_k = E\{x_k x_k^T\}$, which is the second moment of x_k, by maximizing $\lambda_{\min}(P)$ subject to inequality (11) and $P > 0$.

BOUND ON OUTPUT ENERGY FOR UNIT ENERGY INPUTS

Consider system (1) with $x_0 = 0$ with measurement (2) where u_k is a deterministic unit energy input. This time, an upper bound on the second moment of the output is found.

Theorem 4. Let there be a $P > 0$ and $\gamma > 0$

$$\begin{bmatrix} A^T & C^T \\ B^T & D^T \end{bmatrix} \begin{bmatrix} P & O \\ O & I \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix} + \sum_{i=1}^{q} \begin{bmatrix} \left(tr[\Phi'^{i}_{11} P] + tr[\Phi'^{i}_{22}]\right)\Gamma^i & 0 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} P & 0 \\ 0 & \gamma I \end{bmatrix} \leq 0 \tag{19}$$

then for the same initial state and input in theorem 2, the output of the model given by equation (2) satisfies

$$\Sigma_{k=0}^{N} E\{\|y_k\|^2\} \leq \gamma \tag{20}$$

for any $N \geq 0$.

Proof. First, it is shown that inequality (19) for $P > 0$, and $V_k = x_k^T P x_k$ implies that

$$E\{V_{k+1} - V_k\} + E\{\|y_k\|^2\} \leq \gamma \|u_k\|^2 \tag{21}$$

which, then, yields

$$\Sigma_{k=0}^{N} E\{\|y_k\|^2\} \leq \gamma \Sigma_{k=0}^{N} \|u_k\|^2 - E\{V_{N+1}\} \tag{22}$$

that results in (20).
The scalar parameter γ can be minimized subject to $P > 0$ and (19) to get a better bound for the output energy. For the stochastic inputs u_k satisfying (15), the following result is true:
Theorem 5: Under the same conditions as theorem 4, inequality (20) is true for stochastic inputs u_k satisfying (15).
Proof. This time, it is shown that (19) implies

$$E\{V_{k+1}/x_k, u_k\} - V_k + E\{\|y_k\|^2/x_k, u_k\} \leq \gamma \|u_k\|^2 \tag{23}$$

and, so

$$E\{V_{k+1} - V_k\} + E\{\|y_k\|^2\} \leq \gamma E\{\|u_k\|^2\} \tag{24}$$

which yields the result.

BOUND ON OUTPUT ENERGY FOR GIVEN INITIAL STATES

Consider model (1) and (2) with $u_k \equiv 0$ for all $k \geq 0$ and given initial state x_o. Then, we have the following result:
Theorem 6. Let there be a $P > 0$ such that the LMI below holds:

$$P - A^T P A - \Sigma_{i=1}^{q} tr[\Phi'^{i}_{11} P]\Gamma^i - \Sigma_{i=1}^{q} tr[\Phi'^{i}_{11}]\Gamma^i - C^T C \geq 0 \tag{25}$$

then model (1) and (2) with $u_k \equiv 0,\ k \geq 0$ has the following output energy bound

$$\Sigma_{k=0}^{N} E\|y_k\|^2 \leq x_o^T P x_o \tag{26}$$

for all $N \geq 0$.
Proof. It is first shown that the existence of a $P > 0$ that satisfies LMI (25) implies for $V_k = x_k^T P x_k$,

$$E\{V_{k+1}/x_k\} - V_k \leq -E\{\|y_k\|^2/x_k\}$$

from which we obtain

$$\Sigma_{k=0}^{N} E\{V_{k+1} - V_k\} \le -\Sigma_{k=0}^{N} E\{\|y_k\|^2\}$$

$$E\{V_{N+1}\} - x_o^T P x_o \le -\Sigma_{k=0}^{N} E\{\|y_k\|^2\}$$

which implies the result.

The above result can be obtained for random initial states too, in which case, we have

$$\Sigma_{k=0}^{N} E\|y_k\|^2 \le E\{x_o^T P x_o\} \tag{27}$$

for any $N \ge 0$ replacing (26), with the condition (25) remaining the same.

STATE FEEDBACK STABILIZABILITY

Let us consider system (1) again with u_k being the control input. It is assumed that the whole state vector is available for measurement and the control is formed as

$$u_k = Kx_k \tag{28}$$

to stabilize system (1) in the m.s. sense. It follows from inequality (8) that the necessary and sufficient condition for the m.s. stabilizability of system (1) by state feedback is the existence of a $Q > 0$ and K such that

$$Q - (A+BK)Q(A+BK)^T - \Sigma_{i=1}^{q} \Phi_{11}^{i} tr[Q\Gamma^i] > 0 \tag{29}$$

Upon a change of variable $Y = KQ$, (29) yields

$$Q - (AQ+BY)Q^{-1}(AQ+BY)^T - \Sigma_{i=1}^{q} \Phi_{11}^{i} tr[Q\Gamma^i] > 0 \tag{30}$$

which is equivalent to the LMI

$$\begin{bmatrix} Q - \Sigma_{i=1}^{q} \Phi_{11}^{i} tr[Q\Gamma^i] & AQ + BY \\ QA^T + Y^T B^T & Q \end{bmatrix} > 0 \tag{31}$$

Therefore, we have the following result:

Theorem 7. System (1) is m.s. stabilizable by the control (28) if and only if there exist Q and Y such that the LMI (31) holds, in which case $K = YQ^{-1}$.

It is possible to obtain simpler forms of the LMI (31) by using the matrix elimination and Finsler's lemma [6] as follows.

Theorem 8. The result of theorem 7 holds if and only if there exists $Q > 0$ and $\sigma \in \mathbb{R}$ such that

$$\begin{bmatrix} Q - \Sigma_{i=1}^{q} \Phi_{11}^{i} \, tr[Q\Gamma^{i}] - \sigma B B^{T} & AQ \\ QA^{T} & Q \end{bmatrix} > 0 \tag{32}$$

or the equivalent condition for the matrix $Q > 0$ and the scalar $\sigma > 0$ is

$$Q - \Sigma_{i=1}^{q} \Phi_{11}^{i} \, tr[Q\Gamma^{i}] - \sigma BB^{T} - AQA^{T} > 0 \tag{33}$$

If one of LMIs (32) or (33) holds for $Q > 0$, then all the m.s. stabilizing feedback gains are given by

$$K = B^{\dagger}(LQ^{-1/2} - A) + (I - B^{\dagger}B)Z \tag{34}$$

where Z is arbitrary and L is any nxn factor of $(\sigma BB^{T} + AQA^{T})$.

The parametrization of the m.s. stabilizing feedback gains in (34) is found by matching the left-hand sides of (29) and (33) and solving for the required K.

CONTROL FOR STATE BOUND ASSIGNMENT WITH UNIT ENERGY DISTURBANCE

Consider the system below:

$$x_{k+1} = Ax_{k} + Bu_{k} + f_{k} + Hw_{k} \tag{35}$$

where $x_{o} = 0$, u_{k} is the state feedback control given by (28), and w_{k} is a unit energy disturbance. Substituting from (28), we obtain

$$x_{k+1} = (A + BK)x_{k} + f_{k} + Hw_{k} \tag{36}$$

To be able to assign a second moment bound on the state as is given in theorem 2, in the presence of unit energy disturbance w_{k}, we need to have a P > 0 such that

$$\begin{bmatrix} (A+BK)^{T}P(A+BK) + \Sigma_{i=1}^{q} \Gamma^{i} \, tr[\Phi_{11}^{i} P] & (A+BK)^{T} PH \\ H^{T}P(A+BK) & H^{T}PH \end{bmatrix} - \begin{bmatrix} P & 0 \\ 0 & I \end{bmatrix} \leq 0 \tag{37}$$

which simply follows from LMI (11). Using the Schur complement result that

$$\begin{bmatrix} A & B \\ B^{T} & C \end{bmatrix} \geq 0 \tag{38}$$

for C > 0 if and only $A - BC^{-1}B^{T} \geq 0$, (37) yields

$$\begin{bmatrix} P - \Sigma_{i=1}^{q} \Gamma^i tr[\Phi^i_{11} P] & -(A+BK)^T PH & (A+BK)^T P \\ -H^T P(A+BK) & I - H^T PH & 0 \\ P(A+BK) & 0 & P \end{bmatrix} \geq 0 \qquad (39)$$

Eliminating the control gain K, we obtain

$$\begin{bmatrix} P - \Sigma_{i=1}^{q} \Gamma^i tr[\Phi^i_{11} P] & -A^T PH & A^T P \\ -H^T PA & I - H^T PH & 0 \\ PA & 0 & P \end{bmatrix} - \sigma \begin{bmatrix} 0 \\ H^T PB \\ -PB \end{bmatrix} [0 \;\; B^T PH \;\; -B^T P] \geq 0 \qquad (40)$$

for some $\sigma \in \mathbb{R}$. Finally, using the Schur complement result again, (41) yields

$$\begin{bmatrix} P - \Sigma_{i=1}^{q} \Gamma^i tr[\Phi^i_{11} P] & -A^T PH & A^T P & 0 \\ -H^T PA & I - H^T PH & 0 & H^T PB \\ PA & 0 & P & -PB \\ 0 & B^T PH & -B^T P & \sigma^{-1} I \end{bmatrix} \geq 0 \qquad (41)$$

which is an LMI in $P > 0$ and $\sigma^{-1} > 0$. Therefore, to summarize this development, we have Theorem 9. A second moment bound on the state given by Theorem 2 is assignable by state feedback (28) for $x_o = 0$ and unit energy disturbances w_k if the LMI (41) holds for $P > 0$, $\gamma^{-1} > 0$. In case such P and σ^{-1} can be found, the necessary control gain K can be found by substituting these values in LMI (39) and solving the resulting LMI for K.

The above results also applies to the case where w_k is stochastic in nature.

CONTROL FOR BOUND ASSIGNMENT ON OUTPUT ENERGY WITH UNIT ENERGY DISTURBANCE

Consider the state-feedback-controlled system (36) with the same initial condition and the disturbance input. Assume that the regulated output of the system is given by

$$y_k = Cx_k + Dw_k + g_k \qquad (42)$$

Using theorem 4, we should have $P > 0$ and $\gamma > 0$ such that

$$\begin{bmatrix} (A+BK)^T & C^T \\ H^T & D^T \end{bmatrix} \begin{bmatrix} P & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} (A+BK) & H \\ C & D \end{bmatrix} + \sum_{i=1}^{q} \begin{bmatrix} (tr[\Phi^i_{11} P] + tr[\Phi^i_{22}])\Gamma^i & 0 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} P & 0 \\ 0 & \gamma I \end{bmatrix} \leq 0 \qquad (43)$$

This inequality is re-written as

$$\bar{\mathcal{P}} - (A + BK\xi)^T \mathcal{P} (A + BK\xi) \geq 0 \qquad (44)$$

for $\mathcal{P} > 0$, with obvious definitions of the new variables. Using the Schur complement result, (44) is equivalent to

$$\begin{bmatrix} \overline{\mathcal{P}} & (A + BK\xi)^T\mathcal{P} \\ \mathcal{P}(A + BK\xi) & \mathcal{P} \end{bmatrix} \geq 0 \tag{45}$$

Eliminating K in (45) yields

$$\begin{bmatrix} \overline{\mathcal{P}} & A^T\mathcal{P} \\ \mathcal{P}A & \mathcal{P} - \sigma\mathcal{P}BB^T\mathcal{P} \end{bmatrix} \geq 0 \tag{46}$$

or

$$\begin{bmatrix} \overline{\mathcal{P}} & A^T\mathcal{P} & 0 \\ \mathcal{P}A & \mathcal{P} & \mathcal{P}B \\ 0 & B^T\mathcal{P} & \sigma^{-1}I \end{bmatrix} \geq 0 \tag{47}$$

which is an LMI in $P > 0$ and $\sigma^{-1} > 0$.

Theorem 10. A bound on the output energy given by theorem 4 is assignable by state feedback (28) with $x_o = 0$ and unit energy disturbances of the LMI (47) holds with $P > 0$ and $\sigma > 0$. In case the LMI holds, then the necessary control gain can be found by substituting P into (45) and solving the resulting LMI for K.

CONTROL FOR OUTPUT ENERGY BOUND ASSIGNMENT FOR GIVEN INITIAL STATE

Consider the controlled system (36) and the output equation (42) with $w_k \equiv 0,\ k \geq 0$ and given initial condition x_0. To accomplish such a bound on the output energy, it should be true that

$$P - \Sigma_{i=1}^{q}(tr[\Phi'^{i}_{11}P] + tr[\Phi'^{i}_{22}])\Gamma^i - C^TC - (A + BK)^T P(A + BK) \geq 0 \tag{48}$$

for $P > 0$ and K. But (48) is equivalent to

$$\begin{bmatrix} P - \Sigma_{i=1}^{q}(tr[\Phi'^{i}_{11}P] + tr[\Phi'^{i}_{22}])\Gamma^i - C^TC & (A + BK)^TP \\ P(A + BK) & P \end{bmatrix} \geq 0 \tag{49}$$

Eliminating K, we obtain

$$\begin{bmatrix} P - \Sigma_{i=1}^{q}(tr[\Phi'^{i}_{11}P] + tr[\Phi'^{i}_{22}])\Gamma^i - C^TC & A^TP \\ PA & P - \sigma PBB^TP \end{bmatrix} \geq 0 \tag{50}$$

for some $\sigma \in \mathbb{R}$. The Schur complement gives

$$\begin{bmatrix} P - \Sigma_{i=1}^{q} tr[\Phi_{11}^{i} P] + tr[\Phi_{22}^{i}])\Gamma^{i} - C^{T}C & A^{T}P & 0 \\ PA & P & PB \\ 0 & B^{T}P & \sigma^{-1}I \end{bmatrix} \geq 0 \tag{51}$$

which is an LMI in P > 0 and $\sigma^{-1} > 0$.

Theorem 11. A bound on the output energy for a given initial state x_0 is assignable by state feedback (28) in case of no external disturbances of the LMI (51) holds for P > 0 and $\sigma^{-1} > 0$, in which case, the necessary control gain K can be computed by substituting this P > 0 into (49) and solving the resulting LMI for K.

CONCLUSION

We have considered various important analysis and control synthesis problems for a class of discrete-time nonlinear stochastic systems and given conditions for feasibility and problem solution in terms of LMIs. This generalizes the applicability of the LMI approach to an important class of models. Actually, it is possible to generalize these results further by replacing equation (4) by the inequality

$$\left\{ \begin{bmatrix} f_k \\ g_k \end{bmatrix} \begin{bmatrix} f_k^{T} & g_k^{T} \end{bmatrix} \middle| x_k \right\} \leq \Sigma_{i=1}^{q} \begin{bmatrix} \Phi_{11}^{i} & \Phi_{12}^{i} \\ (\Phi_{12}^{i})^{T} & \Phi_{22}^{i} \end{bmatrix} x_k^{T} \Gamma^{i} x_k \tag{52}$$

With this formulation, as mentioned in [4], one can, e.g., include sector-bounded nonlinearities multiplying white noise sequences, such as $f_k = \psi(x_k)v_k$ where $\| \psi(x) \| \leq C \; \|x\|$ and v_k is a scalar white noise sequence. Or, for example, another application possibility is for bilinear terms multiplied by white noise such as $f_k = (x_k^{(1)}x_k^{(2)} + x_k^{(1)}x_k^{(3)} + \ldots + x_k^{(2)}x_k^{(3)} + \ldots + x_k^{(n)}x_k^{(n-1)})v_k$ where $x_k^{(i)}$ is the ith component of the state vector and v_k is a zero mean white noise vector. This f_k can easily be shown to satisfy (52).

REFERENCES

[1] D. H. Jacobson, "A general result in stochastic optimal control of nonlinear discrete-time systems with quadratic performance criteria," J. Math. Anal. Appl., vol. 47, pp. 153-161, 1974.

[2] E. Yaz, "Linear state estimators for nonlinear stochastic systems with noisy nonlinear observations," Int. J. Control, vol. 48, pp. 2465-2475, 1988.

[3] E. Yaz, "Infinite horizon quadratic optimal control of a class of nonlinear stochastic systems," IEEE Trans. Autom. Control, vol. 34, pp. 1176-1180, 1989.

[4] E. Yaz, "Robust design of stochastic controllers for nonlinear systems," IEEE Trans. Autom. Control, vol. 34, pp. 349-353, 1989.

[5] T. Morozan, "Optimal stationary control for dynamic systems with Markov perturbations," Stoc. Analysis and Applic.s, vol. 1, pp. 299-325, 1983.

[6] S. Boyd, L. El Ghaoui, E. Feron, V. Balakrishnan, Linear Matrix Inequalities in System and Control Theory, SIAM, Philadelphia, PA, 1994.

Robustness and $\mathcal{L}^\infty$ Performance of Nonlinear Feedback Systems *

Ian J. Fialho and Tryphon T. Georgiou
Department of Electrical Engineering
University of Minnesota
Minneapolis, Minnesota 55455

Abstract

A framework for robustness analysis of nonlinear systems is overviewed. Evaluation of the robustness margins as well as performance considerations motivate the development of computational tools for assessing the $\mathcal{L}^\infty$-gain of nonlinear systems. These are developed in a variational framework and rely on the theory of viscosity solutions of Hamilton Jacobi equations. The theory is illustrated on a saturated feedback loop example.

1 Introduction

In this paper we motivate and develop computational tools for $\mathcal{L}^\infty$-gain analysis of nonlinear feedback systems. To this end we review a suitable nonlinear generalization of the gap metric, given by Georgiou and Smith [9, 10], which serves to quantify the unstructured uncertainty that can be tolerated in a feedback loop while stability is maintained. The stability margin of the feedback loop is related to the $\mathcal{L}^\infty$-gain of the parallel projection operator (see [6, 9, 10]). Independently, the $\mathcal{L}^\infty$-gain of closed loop operators serves to quantify performance of the feedback loop. These considerations suggest the need for computational tools for $\mathcal{L}^\infty$-gain analysis of nonlinear feedback systems. The development of such techniques is the main focus of this paper.

The present framework relies heavily on the recently developed theory of viscosity solutions to nonlinear Hamilton Jacobi equations (see [4, 5]). We pose the problem of assessing the $\mathcal{L}^\infty$-gain in a variational framework and approach it via viscosity solutions of an associated variational inequality.

Analogous techniques for computation of $\mathcal{L}^2$-gains have received considerable attention in the literature (see [12, 14]) and have a close connection with the theory of dissipative dynamical systems [15, 11]. The current framework is quite similar and parallels the $\mathcal{L}^2$ theory. Earlier work which is similar in spirit includes [1] and [2]. The reader is referred to [9, 10, 7] for compete proofs and a more detailed exposition.

---

*Supported by the NSF under Grant NSF/ECS-9016050 and the AFOSR under Grant AF/F49620-92-J-0241.

2 Robustness/Performance of Nonlinear Feedback Systems

In this section we motivate the need to assess the $\mathcal{L}^\infty$-gain of general nonlinear systems. To this end consider the following feedback interconnection where $\mathbf{P}$ denotes a given nominal system and $\mathbf{C}$ denotes a feedback controller. These are typically described via dynamical state-equations, e.g., $\dot{x}(t) = f(x,u),\ y(t) = h(x,u)$. For the purposes of this discussion we will consider systems to be causal, nonlinear operators defined on extended spaces, i.e., $\mathcal{U} = \mathcal{L}_e^\infty[0,\infty)$ and $\mathcal{Y} = \mathcal{L}_e^\infty[0,\infty)$, $\mathbf{P} : \mathcal{U} \to \mathcal{Y}$ and $\mathbf{C} : \mathcal{Y} \to \mathcal{U}$ respectively. The goal of feedback stabilization is to ensure that the mapping from $(u_0,\ y_0) \to (u_1,\ u_2,\ y_1,\ y_2)$ is stable, i.e., the induced $\mathcal{L}^\infty$-norm is bounded.

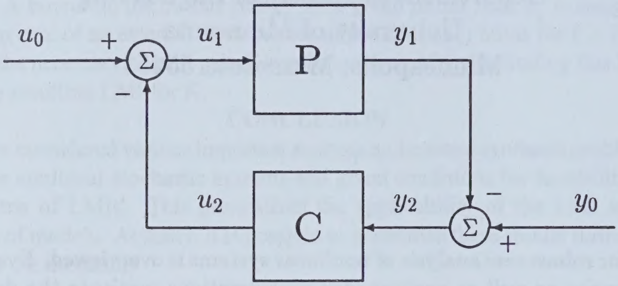


Figure 1: Standard feedback configuration $[\mathbf{P},\mathbf{C}]$.

The *graph* of a system $\mathbf{P}$ is defined to be $\mathcal{G}_\mathbf{P} := \{(u,\ \mathbf{P}u) : u \in \mathcal{U},\ \mathbf{P}u \in \mathcal{Y}\}$, and the *inverse graph* of $\mathbf{C}$, $\mathcal{G}'_\mathbf{C} := \{(\mathbf{C}y,\ y) : y \in \mathcal{Y},\ \mathbf{C}y \in \mathcal{U}\}$. The following notion of distance, defined in [9, 10], can be used to quantify system perturbations, and thereby, unstructured uncertainty and modeling error. Here $\mathcal{S}_\alpha$ denotes the open ball of radius α.

$$\vec{\delta}_{\mathcal{S}_\alpha}(\mathcal{G}_\mathbf{P},\mathcal{G}_{\mathbf{P}_1}) \ := \ \begin{cases} \inf\{\|(\mathbf{\Phi} - \mathbf{I})|_{\mathcal{G}_\mathbf{P}\cap\mathcal{S}_\alpha}\| : \mathbf{\Phi} \text{ is a continuous causal operator} \\ \text{which maps } \mathcal{G}_\mathbf{P}\cap\mathcal{S}_\alpha \text{ into } \mathcal{G}_{\mathbf{P}_1} \text{ injectively,} \\ \text{and is such that } \mathbf{T}_\tau(\mathbf{I}-\mathbf{\Phi})\mathbf{T}_\tau|_{\mathcal{G}_\mathbf{P}\cap\mathcal{S}_\alpha} \text{ is compact}\}\,, \\ \\ \infty \text{ if no such operator } \mathbf{\Phi} \text{ exists.} \end{cases}$$

The following proposition gives the robustness margin as the inverse of the norm of the *parallel projection operator* (onto $\mathcal{G}_\mathbf{P}$ parallel to $\mathcal{G}'_\mathbf{C}$). This operator is the closed loop mapping $\mathbf{\Pi}_{\mathcal{G}_\mathbf{P}//\mathcal{G}'_\mathbf{C}} : (u_0,\ y_0) \to (u_1,\ y_1)$. Below, for any mapping $\mathbf{A}$, we define $\|\mathbf{A}\| := \sup_{\tau,\|\mathbf{T}_\tau \mathbf{w}\|\neq 0} \frac{\|\mathbf{T}_\tau \mathbf{A}\mathbf{w}\|}{\|\mathbf{T}_\tau \mathbf{w}\|}$ where $\mathbf{T}_\tau$ denotes the truncation operator onto the interval $[0,T]$.

Theorem. [9, 10] Let $[\mathbf{P},\mathbf{C}]$ be bounded on $\mathcal{S}_\alpha$ with $\|\mathbf{\Pi}_{\mathcal{G}_\mathbf{P}//\mathcal{G}'_\mathbf{C}}|_{\mathcal{S}_\alpha}\| = \beta$, and let $\mathbf{P}_1$ be such that $\vec{\delta}_{\mathcal{S}_{\beta\alpha}}(\mathcal{G}_\mathbf{P},\mathcal{G}_{\mathbf{P}_1}) = \gamma$ with $\gamma < \beta^{-1}$. Then $[\mathbf{P}_1,\mathbf{C}]$ is bounded on $\mathcal{S}_{\alpha(1-\beta\gamma)}$ and moreover

$$\|\mathbf{\Pi}_{\mathcal{G}_{\mathbf{P}_1}//\mathcal{G}'_\mathbf{C}}|_{\mathcal{S}_{\alpha(1-\beta\gamma)}}\| \leq \frac{1+\gamma}{1-\beta\gamma}\|\mathbf{\Pi}_{\mathcal{G}_\mathbf{P}//\mathcal{G}'_\mathbf{C}}|_{\mathcal{S}_\alpha}\|.$$

□

In addition to providing a robustness margin in the above framework, the $\mathcal{L}^\infty$-gain often serves as an accurate measure of performance. In the following sections we develop the necessary computational tools, for cases where the relevant mapping (e.g., $\Pi_{\mathcal{G}_P//\mathcal{G}'_C}$ above) can be represented by a finite-dimensional nonlinear dynamical system. More specifically we consider nonlinear systems Σ characterized by

$$\begin{aligned} \dot{x}(t) &= f(t, x(t), u(t)) \\ y(t) &= h(t, x(t), u(t)), \end{aligned} \tag{1}$$

with $t \in [0, T]$, $x \in \mathbf{R}^n$, $y \in \mathbf{R}^p$, and $u \in \mathbf{R}^m$. We focus on computing the performance/value function[1]

$$V_a(t, x) = \sup_{u \in \mathcal{U}[t,T], x(t)=x} \left\{ \text{ess.sup}_{t \le \tau \le T} |h(\tau, x(\tau), u(\tau))| \right\}, \tag{2}$$

subject to the dynamics specified in (1). Then the $\mathcal{L}^\infty$-gain (over $[t, T]$), starting from the initial condition $x(t) = x$, is given by $\|\Sigma|_{S_\alpha}\| = \sup_{a \in (0,\alpha]} \frac{V_a(t,x)}{a}$.

3 $\mathcal{L}^\infty$-Gain Analysis of Nonlinear Systems

In this section we characterize $V_a(t, x)$ as a viscosity solution of an associated variational inequality. For notational convenience we drop the subscript a. We begin with the finite horizon ($T < \infty$) case. The following assumption guarantees that the dynamical equation (1) has a unique solution and that the value function V is continuous:

$$f, h \text{ are continuous, } f(t, 0, 0) = 0 \ (t \in [0, T]), \text{and } f \text{ is globally Lipschitz in } x \text{ uniformly in } t \text{ and } u. \tag{3}$$

We remark that the global Lipschitz assumption may be relaxed to a local Lipschitz assumption if one is interested in evaluating V on a bounded, invariant subset of $\mathbf{R}^n$. However, for ease of presentation we work on $\mathbf{R}^n$.

The following evolutionary variational inequality (EVI) is of interest:

$$\max \left\{ \max_{u \in \mathrm{U}} |h(t, x, u)| - v(t, x), \frac{\partial v}{\partial t}(t, x) + \max_{u \in \mathrm{U}} \frac{\partial v}{\partial x}(t, x) \cdot f(t, x, u) \right\} = 0. \tag{4}$$

The notion of a viscosity solution of (4) is presented below. Here $v_*(v^*)$ denotes the lower(upper) semicontinuous envelope of v.

Definition 1. ([5, 1, 8]) A locally bounded function v on $(0, T) \times \mathbf{R}^n$, is said to be a *viscosity solution* of (4) if the following two conditions hold.

(a) For any $\varphi \in C^\infty((0, T) \times \mathbf{R}^n)$ such that $v_* - \varphi$ has a local minimum at $(t_0, x_0) \in (0, T) \times \mathbf{R}^n$,

$$\max \left\{ \max_{u \in \mathrm{U}} |h(t_0, x_0, u)| - v_*(t_0, x_0), \frac{\partial \varphi}{\partial t}(t_0, x_0) + \max_{u \in \mathrm{U}} \frac{\partial \varphi}{\partial x}(t_0, x_0) \cdot f(t_0, x_0, u) \right\} \le 0.$$

---

[1] $\mathcal{U}[t, T]$ denotes the class of measurable input functions from $[t, T] \to \mathrm{U} = \{u \in \mathbf{R}^m : |u| \le a\}$

(In this case v_* is said to be a viscosity supersolution.)

(b) For any $\varphi \in C^\infty((0,T) \times \mathbf{R}^n)$ such that $v^* - \varphi$ has a local maximum at $(t_0, x_0) \in (0,T) \times \mathbf{R}^n$,

$$\max\left\{\max_{u \in \mathrm{U}} |h(t_0, x_0, u)| - v^*(t_0, x_0), \frac{\partial \varphi}{\partial t}(t_0, x_0) + \max_{u \in \mathrm{U}} \frac{\partial \varphi}{\partial x}(t_0, x_0) \cdot f(t_0, x_0, u)\right\} \geq 0.$$

(In this case v^* is said to be a viscosity subsolution.)

Theorem 1. The value function V is a viscosity solution of the Bellman equation (4) on $(0,T) \times \mathbf{R}^n$, with boundary condition $V(T,x) = \max_{u \in \mathrm{U}} |h(T,x,u)|$. □

The comparison/uniqueness results for (4) developed in [7] establish that V is actually the unique viscosity solution of (4) with boundary condition $V(T,x) = \max_{u \in \mathrm{U}} |h(T,x,u)|$. They also play a role in the analysis of the infinite horizon case and in establishing the convergence of numerical approximation schemes.

We now turn our attention to the infinite horizon ($T \to \infty$) case and consider time-invariant nonlinear systems described by

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)), & (5)\\ y(t) &= h(x(t)), & (6) \end{aligned}$$

for $t \in [0, \infty)$. The following assumptions guarantee existence and uniqueness of solutions over the infinite interval:

$$f \in C(\mathbf{R}^n \times \mathrm{U}),\ h \in C(\mathbf{R}^n),\ f(0,0) = 0, f \text{ is globally Lipschitz in } x \text{ uniformly for } u \in \mathrm{U}. \qquad (7)$$

We consider the value function

$$V(x) = \sup_{u \in \mathcal{U}[0,\infty),\, x(0)=x} \left\{\text{ess.sup}_{0 \leq \tau < \infty} |h(x(\tau))|\right\}, \qquad (8)$$

subject to the dynamics (5). Unlike the finite horizon case, the value function in (8) is merely lower semicontinuous on $\mathbf{R}^n$. The lower semicontinuity assertion is the best possible. An example which demonstrates this case is: $\dot{x} = -x + 2\sin(x)u$, $y = x$, and $-1 \leq u \leq 1$. In this case $V(0) = 0$ whereas $V(x) = 1.8955$ for $0 < |x| \leq 1.8955$ ($V(x) = x$ for larger values of $|x|$).

The following stationary variational inequality (SVI) is of interest:

$$\max\left\{|h(x)| - v(x)\ ,\ \max_{u \in \mathrm{U}} \frac{\partial v}{\partial x}(x) \cdot f(x,u)\right\} = 0. \qquad (9)$$

Theorem 2. If the infinite horizon value function V is locally bounded on $\mathbf{R}^n$ then it is a viscosity solution of the stationary Bellman equation (9). □

It turns out that the value function is the minimal lower semicontinuous viscosity solution of the Bellman equation.

Theorem 3. If there exists a locally bounded lower semicontinuous viscosity solution v of (9) such that $v(x) \geq |h(x)|$, $\forall\, x \in \mathbf{R}^n$ then $V(x) \leq v(x)$ for all $x \in \mathbf{R}^n$. □

4 Approximation Schemes and Computational Aspects

We proceed to outline approximation schemes for the solution of the Bellman equations (4) and (9), based on discrete dynamic programming. We first consider the finite horizon case. Let $T > 0$ and define the Euler discretization of (1)

$$\begin{aligned} x((j+1)\Delta) &= x(j\Delta) + \Delta \cdot f(j\Delta, x(j\Delta), u(j\Delta)), \\ y(j\Delta) &= h(j\Delta, x(j\Delta), u(j\Delta)), \end{aligned} \tag{10}$$

with j, N integers, $0 \le j \le N$, $N\Delta = T$, and $u(j\Delta) \in \mathrm{U}$. Define the value function,

$$\mathrm{V}_\Delta(j\Delta, x) = \sup_{u \in \mathcal{U}_\Delta[j\Delta, N\Delta], x(j\Delta)=x} \left\{ \sup_{j \le k \le N} |h(k\Delta, x(k\Delta), u(k\Delta))| \right\} \tag{11}$$

subject to (10). Here $\mathcal{U}_\Delta[j\Delta, N\Delta]$ denotes the class of input functions that are piecewise constant on the intervals $[j\Delta, (j+1)\Delta)$. For $0 \le j \le N-1$, V_Δ satisfies the dynamic programming difference equation:

$$\mathrm{V}_\Delta(j\Delta, x) = \max_{u \in \mathrm{U}} \{\max\{\mathrm{V}_\Delta((j+1)\Delta, x + \Delta f(x,u)), |h(j\Delta, x, u)|\}\} \tag{12}$$

with $\mathrm{V}_\Delta(N\Delta, x) = \max_{u \in \mathrm{U}} |h(N\Delta, x, u)|$. Using the discrete dynamic programming property we establish that, for small Δ, V_Δ represents a close approximation of $V(t, x)$ on the interval $[0, T]$. Define

$$\mathrm{V}^*(t, x) = \lim_{\Delta \to 0} \sup\{\mathrm{V}_\delta(j\delta, y) : \delta, j, y \text{ such that } \delta \le \Delta, |j\delta - t| < \Delta, |y - x| < \Delta\},$$

and

$$\mathrm{V}_*(t, x) = \lim_{\Delta \to 0} \inf\{\mathrm{V}_\delta(j\delta, y) : \delta, j, y \text{ such that } \delta \le \Delta, |j\delta - t| < \Delta, |y - x| < \Delta\}.$$

We have the following approximation result.

Theorem 4. Let V_Δ satisfy (12). Then $V(t, x) = \mathrm{V}^*(t, x) = \mathrm{V}_*(t, x)$. □

The infinite horizon version of Theorem 4 is as follows.

Theorem 5. Assume that $\sup_{\Delta \le \Delta_0} \mathrm{V}_\Delta(x)$ is locally bounded. Then the following hold:

(a) $V(x) \le \mathrm{V}_*(x)$, i.e., $V(x)$ is locally bounded.

(b) If $\lim_{x \to \infty} |h(x)| = \infty$ then $V(x) = \mathrm{V}_*(x)$. □

In order to compute the infinite horizon value function we would, in principle, have to integrate (4) backward in time, for larger and larger T. In many cases it would be sufficient to compute only $V(0)$, and merely bound the value function outside the set of states reachable from the origin. The control theoretic origin of (4) and (9) may then be exploited to yield a scheme for the localization of the solution $V(x)$.

We assume that the origin is an attractive equilibrium. Define, $h_\gamma(x) = \max\{\gamma, |h(x)|\}$ and consider the following variational inequalities:

$$\max\left\{ h_\gamma(x) - v(x),\ \max_{u \in \mathrm{U}} \frac{\partial v}{\partial x}(x) \cdot f(x, u) \right\} = 0, \tag{13}$$

$$\max\left\{h_\gamma(x) - v(t,x),\ \frac{\partial v}{\partial t}(t,x) + \max_{u\in \mathrm{U}} \frac{\partial v}{\partial x}(t,x)\cdot f(x,u)\right\} = 0, \tag{14}$$

with boundary condition $v(T,x) = h_\gamma(x)$.

Proposition 1. Let $\gamma > 0$. The following are equivalent.

(a) $h_\gamma(x) \geq V(x)$, $\forall\, x \in \mathbf{R}^n$.

(b) $h_\gamma(x)$ is the value function associated with the dynamics (5) and an output function $h(x) = h_\gamma(x)$.

(c) $h_\gamma(x)$ is a viscosity solution of the stationary Bellman equation (13).

(d) $v(t,x) = h_\gamma(x)$ is the unique continuous viscosity solution of the evolutionary Bellman equation (14). □

Proposition 1 suggests an efficient approach to test possible upper bounds for the value function V. Given a value γ, constancy of $\mathrm{V}_\Delta(j\Delta, x)$ through one step of (12) (with $h(j\Delta, x, u) = h_\gamma(x)$) implies that $h_\gamma(x) \geq V(x)$. This approach to compute an upper bound is constructive, unlike standard Lyapunov methods (cf. [13]). We now address the issue of lower bounds for V.

Proposition 2. For $\gamma > 0$ let $v(t,x)$ be the unique viscosity solution of (14). Then

(a) $h_\gamma(0) < V(0)$ if and only if $v(t,0) > h_\gamma(0)$ for some $t < T$ and T large enough.

(b) If $h_\gamma(0) < V(0)$ then $\lim_{T\to\infty} v(0,x) = V(x)$. □

Let $\mathrm{V}_\Delta(j\Delta, x)$ be the solution of (12) with $h(j\Delta, x, u) = h_\gamma(x)$. For small enough $\Delta > 0$, if γ is such that there exists j for which $\mathrm{V}_\Delta(j\Delta, 0) > h_\gamma(0)$, then $V(x) \geq h_\gamma(x)$.

Propositions 1 and 2 suggest ways to obtain upper/lower bound the solution. We point out here that it is possible to accelerate the convergence of the discrete recursion (12). We refer the reader to [3] for further discussion on this and other issues related to propogating recursions such as (12). Finally we present an example that illustrate the ideas developed in this paper.

Example. We consider the saturated feedback loop shown in Figure 2. We outline how to compute the semiglobal $\mathcal{L}^\infty$-induced norm $\|\Sigma|_{S_1}\|$ of the input output operator $\Sigma : u \to y$.

In state space form this mapping has the following representation,

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \mathrm{sat}(u - x_1 - x_2) \\ y &= x_1 + x_2. \end{aligned}$$

We proceed to solve the associated variational inequality. As a first step we evaluate $V(x)$ for $|u| \leq 1$. The discrete approximation scheme for the evolutionary variational inequality is

$$\mathrm{V}_\Delta(j\Delta, x) = \max\left\{|x_1 + x_2|,\ \max_{|u|\leq 1} \mathrm{V}_\Delta((j+1)\Delta, (x_1 + \Delta x_2,\ x_2 + \Delta \mathrm{sat}(u - x_1 - x_2))')\right\}$$

with $V_\Delta(N\Delta, x) = |x_1 + x_2|$.

We propagate $V_\Delta(j\Delta, x)$ for $j = N, N-1, \ldots,$ to obtain a steady state solution $V_\Delta(x)$. For small enough Δ, $V_\Delta(x)$ is an approximation to $V(x)$ (see Section 4). Figure 3 shows the level surfaces of $V_\Delta(x)$. The value at the origin is $V_\Delta(0) = 1.7098$. The computations were carried out with $\Delta = 0.05$, using a 600×600 grid. The result after 400 iterations is shown in Figures 3 and 4. The innermost level curve in Figure 3 is the set of states reachable from the origin. It is not too difficult to justify the shape of the level curves in Figure 3. The reader is referred to [7] for details. A plot of $\frac{V_a(0)}{a}$ for $a \in (0, 1]$ is shown in Figure 5. The saturation of this curve as $a \to 0$ corresponds to the feedback loop operating in the linear regime, i.e., a's for which the saturation is inactive. We conclude that $\|\Sigma|_{S_1}\| = 1.73$. □

5 Conclusions

In this paper we highlight the importance of $\mathcal{L}^\infty$-gain computations in the robustness and performance analysis of nonlinear feedback systems. The $\mathcal{L}^\infty$-gain of nonlinear systems is characterized by means of the value function of a variational problem. This value function is shown to be a viscosity solution of an associated variational inequality. Computation of the viscosity solution is developed in the framework of discrete dynamic programming. The numerical aspects of the paper are demonstrated on a representative example.

References

[1] E. N. Barron and H. Ishii, The Bellman Equation for Minimizing the Maximum Cost, *Nonlinear Analysis, Theory, Methods and Applications*, vol. 13, no. 9, pp. 1067-1090.

[2] E. N. Barron, Differential Games with Maximum Cost, *Nonlinear Analysis, Theory Methods and Applications*, vol. 14, no. 11, pp. 971-989, 1990.

[3] I. Capuzzo Dolcetta and M. Falcone, Discrete Dynamic Programming and Viscosity Solutions of the Bellman Equation, *Ann. Inst. H. Poincare, Anal. Non Lin*, vol. 6 (supplement), pp. 161-183, 1989.

[4] M. G. Crandall, L. C. Evans and P. L. Lions, Some Properties of Viscosity Solutions of Hamilton-Jacobi Equations, *Trans. of the American Mathematical Society*, vol. 282, no. 2, pp. 487-502, 1984.

[5] M. G. Crandall, H. Ishii and P. L. Lions, User's Guide to Viscosity Solutions of Second Order Partial Differential Equations, *Bull. of the American Mathematical Society*, vol.27, no. 1, pp. 1-67, 1992.

[6] J.C. Doyle, T.T. Georgiou, and M.C. Smith, The parallel projection operators of a nonlinear feedback system, *Systems & Control Letters*, **20**, 79–85, 1993.

[7] I. J. Fialho and T. T. Georgiou, $\mathcal{L}^\infty$-Gain Analysis of Nonlinear Systems, *submitted to IEEE Transactions on Automatic Control*, June 1995.

[8] W. H. Fleming and H. M. Soner, *Controlled Markov Processes and Viscosity Solutions*, Springer-Verlag, Applications of Mathematics, vol. 25, 1993.

[9] T. T. Georgiou and M. C. Smith, Remarks on Robustness for Nonlinear Feedback Systems, *Proc. 34th IEEE Conference on Decision and Control*, New Orleans, 1995.

[10] T. T. Georgiou and M. C. Smith, Distance Measures for Uncertain Nonlinear Systems, *Proc. European Control Conference*, Rome, 1995.

[11] M. R. James, A partial differential inequality for dissipative nonlinear systems, *Systems and Control Letters*, vol. 21, pp. 315-320, 1993.

[12] M. R. James, Computing the $\mathcal{H}_\infty$ Norm for Nonlinear Systems, *Proc. 12th IFAC World Congress, Sydney*, 1993.

[13] H. K. Khalil, *Nonlinear Systems*, Macmillan Publishing Company, 1992.

[14] A. J. van der Schaft, $\mathcal{L}_2$-Gain Analysis of Nonlinear Systems and Nonlinear State Feedback H_∞ Control, *IEEE Transactions on Automatic Control*, vol. 37, no. 6, pp. 770-784, 1992.

[15] J. C. Willems, Dissipative Dynamical Systems Part I : General Theory, *Archive for Rational Mechanics and Analysis*, vol. 45, pp. 321-351, 1972.

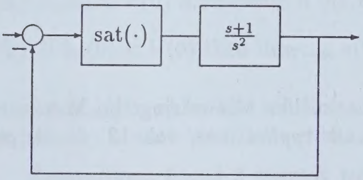


Figure 2: Saturated feedback loop

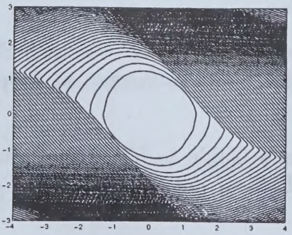


Figure 3: Level curves of $V_\Delta(x)$

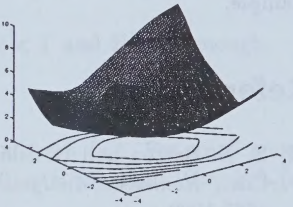


Figure 4: $V_\Delta(x)$

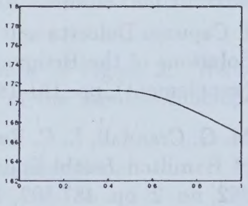


Figure 5: $\frac{V_a(0)}{a}$ versus a

On H^{∞} Optimal Sampled-data Repetitive Controllers[1]

Thaddeus Eldon Peery and *Hitay Özbay*
Department of Electrical Engineering
The Ohio State University
Columbus, OH 43210

Abstract

A definition is given for sampled-data repetitive control design. An H^{∞} optimal sampled-data tracking performance with robust stability problem is formulated using two degree of freedom controllers. Explicit formulas are derived for constructing the discrete-time equivalent problem for general H^{∞} optimal sampled-data problems. A solution is obtained by converting the resulting singular discrete-time equivalent problem to an equivalent continuous-time problem satisfying the standard assumptions. The parameterization of all solutions in terms of the central controller is then used to recover a digital repetitive structure by means of a genetic algorithm.

1 Introduction

Repetitive control is used in numerous industrial applications, see e.g. [10, 16, 18], and references therein. The discrete-time formulation, see e.g. [18], assumes that the plant and performance requirements exist in (or are well approximated in) discrete-time. The new area of sampled-data design addresses continuous-time requirements directly, while designing discrete-time controllers without making any approximations, see e.g. [13, 3]. The extension of this approach to repetitive control is studied here.

In Sections 2-4 we describe the lifting technique and some previous results from [3]. In Section 5 we propose a formal definition for sampled-data repetitive control design and formulate a tracking performance with robust stability problem. In Section 6 we extend the results of [3] to the general case required for our formulation. In Sections 7-9 we present a solution methodology for obtaining sampled-data repetitive controllers, assuming the discrete-time equivalent system can be (has been) calculated. Finally, in Section 10 we give a summary of the paper.

2 Approach to the Sampled-data Problem

We will consider sampled-data control design problems of the form shown in Figure 1, where any required sampling pre-filters have been included in the generalized plant, G. Figure 1 has the structure of the "standard problem", see e.g. [5], where the performance/robustness of the system is characterized by the norm of the transfer function T_{zw}. The extension of standard problem to sampled-data control, see e.g. [3, 13], can be thought of as control design using time-varying continuous-time controllers of the special form $\mathcal{H}_{\tau}C\mathcal{S}_{\tau}$, where C is a discrete-time controller. We assume ideal sampling, fixed zero-order hold, synchronized sample and hold functions operating at the same rate, and zero computational delay. This common approach, see e.g. [19, 6, 4], allows the use of the lifting technique based approach of [3, 2]. The other main approach, a delay of one sample period, is addressed in [13]. Their approach is also based on the lifting technique, but has several other significant differences in addition to the inclusion of a computational delay.

---

[1] This work was supported in part by NSF under grant No. MSS-9203418.

The ultimate goal of the lifting technique is to find an "equivalent" finite dimensional discrete-time system that is norm preserving. Then, the equivalent discrete-time system can be used for norm based discrete-time controller synthesis using the corresponding norm. We are concerned with equivalence in terms of the infinity-norm.

3 The Lifting Technique

The purpose of the lifting technique is to obtain a discrete-time system that represents a sampled continuous-time system without *any* loss of information. Define a lifting operator, W_τ, that piecewise lifts continuous-time signals to discrete-time sequences of time limited functions. The action of W_τ is illustrated in Figure 2 and is defined by

$$W_\tau : L_e^p[0,\infty) \to l_{L^p[0,\tau]}; \quad \hat{f}_i = W_\tau f, \quad \hat{f}_i(t) = f(\tau i + t), \quad 0 \le t < \tau. \tag{1}$$

The following standard notation is used here: $l_{L^p[0,\tau]}$ is the space of one-sided infinite sequences with elements in $L^p[0,\tau]$; and $l_{\mathbb{R}^x}$ is the space of one-sided infinite sequences with elements in $\mathbb{R}^n$, where x is an n-vector. The operator W_τ is invertible, where the inverse operation is defined by $g = W_\tau^{-1}\hat{g}, \quad g(t) = \hat{g}_i(t - \tau i), \quad \tau i \le t < \tau(i+1)$. The inverse lifting operator, W_τ^{-1}, is noncausal. Indeed, if it were defined to be causal, then $W_\tau^{-1}W_\tau$ would be the unit delay operator instead of the identity operator.

By Theorem 4 of [7], the stability of the two systems $\mathcal{F}(\hat{\mathcal{H}}_\tau \hat{G}\hat{\mathcal{S}}_\tau, C)$ and $\mathcal{F}(G, \mathcal{H}_\tau C \mathcal{S}_\tau)$ is equivalent for all non-pathological sampling periods, τ, for all finite dimensional linear time-invariant (LTI) generalized plants, G. The lifted sample and hold operators are

$$\hat{\mathcal{H}}_\tau : \mathbb{R}^{\mathrm{u}} \to \mathrm{L}^{\mathrm{p}}[0,\tau]; \quad \hat{\mathrm{u}}_\mathrm{k} = \hat{\mathcal{H}}_\tau \tilde{\mathrm{u}}_\mathrm{k} \quad \Leftrightarrow \quad \hat{\mathrm{u}}_\mathrm{k}(\mathrm{t}) = \tilde{\mathrm{u}}_\mathrm{k}, \quad 0 \le \mathrm{t} < \tau \text{ and} \tag{2}$$

$$\hat{\mathcal{S}}_\tau : L^p[0,\tau] \to \mathbb{R}^{\mathrm{y}}; \quad \tilde{\mathrm{y}}_\mathrm{k} = \hat{\mathcal{S}}_\tau \hat{\mathrm{y}}_\mathrm{k} \quad \Leftrightarrow \quad \tilde{\mathrm{y}}_\mathrm{k} = \hat{\mathrm{y}}_\mathrm{k}(0). \tag{3}$$

The superscripts indicate the range spaces of the variables, e.g. $\tilde{y}$ denotes a real valued discrete-time variable, while $\hat{y}$ denotes a function valued discrete-time variable. Absorbing $\hat{\mathcal{H}}_\tau$ and $\hat{\mathcal{S}}_\tau$ into the plant we get the system shown in Figure 3, where

$$\tilde{G} = \begin{bmatrix} \hat{G}_{11} & \hat{G}_{12}\hat{\mathcal{H}}_\tau \\ \hat{\mathcal{S}}_\tau \hat{G}_{21} & \hat{\mathcal{S}}_\tau \hat{G}_{22}\hat{\mathcal{H}}_\tau \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \hat{z} \\ \tilde{y} \end{bmatrix} = \tilde{G} \begin{bmatrix} \hat{w} \\ \tilde{u} \end{bmatrix}. \tag{4}$$

The mappings of the blocks of $\tilde{G}$ are as follows: $\tilde{G}_{11} : L^p[0,\tau] \to L^p[0,\tau]$, $\tilde{G}_{12} : l_{\mathbb{R}^u} \to L^p[0,\tau]$, $\tilde{G}_{21} : L^p[0,\tau] \to l_{\mathbb{R}^y}$, and $\tilde{G}_{22} : l_{\mathbb{R}^u} \to l_{\mathbb{R}^y}$. Since $\hat{w} = W_\tau w$ and $\hat{z} = W_\tau z$,

$$\mathcal{F}(\tilde{G}, C) = W_\tau \mathcal{F}(G, \mathcal{H}_\tau C \mathcal{S}_\tau) W_\tau^{-1} \Rightarrow \|\mathcal{F}(\tilde{G}, C)\| = \|\mathcal{F}(G, \mathcal{H}_\tau C \mathcal{S}_\tau)\|. \tag{5}$$

Thus norm based control problems for this discrete-time system are equivalent to the corresponding control problems for the sampled-data system of Figure 1.

4 The Discrete-time Equivalent

For the details of the above development see [3]. The development in the preceding section is completely general in that it is valid for all norms. When the norm is the infinity-norm, the equivalent problem $\tilde{G}$ is in turn equivalent to a finite dimensional problem $\acute{G}$, see [3]. The equivalence is with respect to internal stability and $\|\mathcal{F}(G, \mathcal{H}_\tau C \mathcal{S}_\tau)\|_\infty < 1 \Leftrightarrow \|\mathcal{F}(\acute{G}, C)\|_\infty < 1$. The finite dimensional equivalent system is given by:

$$\acute{G} := \left(\acute{A}, \begin{bmatrix} \acute{B}_1 \\ \acute{B}_2 \end{bmatrix}, \begin{bmatrix} \acute{C}_1 & \acute{C}_2 \end{bmatrix}, \begin{bmatrix} 0 & \acute{D}_{12} \\ 0 & 0 \end{bmatrix} \right), \tag{6}$$

where the real finite dimensional matrices are defined in terms of finite rank operators as follows: $\acute{A} := e^{A\tau} + \hat{B}_1\hat{D}_{11}^* \left[I - \hat{D}_{11}\hat{D}_{11}^*\right]^{-1} \hat{C}_1$,

$$\left[\begin{array}{cc} \acute{C}_1 & \acute{D}_{12} \end{array}\right] := \left[\begin{array}{cc} \Sigma_{CD}^{1/2} & 0 \end{array}\right] T_{CD}, \;\; \acute{B}_2 := \hat{B}_1\hat{D}_{11}^* \left[I - \hat{D}_{11}\hat{D}_{11}^*\right]^{-1} \hat{D}_{12} + \Psi(\tau)B_2,$$

$$\acute{B}_1 := T_B^* \left[\begin{array}{c} \Sigma_B^{1/2} \\ 0 \end{array}\right], \;\; T_B^* \left[\begin{array}{cc} \Sigma_B & 0 \\ 0 & 0 \end{array}\right] T_B := \hat{B}_1 \left[I - \hat{D}_{11}^*\hat{D}_{11}\right]^{-1} \hat{B}_1^*, \;\; \acute{C}_2 := C_2,$$

$$T_{CD}^* \left[\begin{array}{cc} \Sigma_{CD} & 0 \\ 0 & 0 \end{array}\right] T_{CD} := \left[\begin{array}{c} \hat{C}_1^* \\ \hat{D}_{12}^* \end{array}\right] \left[I - \hat{D}_{11}\hat{D}_{11}^*\right]^{-1} \left[\begin{array}{cc} \hat{C}_1 & \hat{D}_{12} \end{array}\right],$$

and $\Psi(t) := \int_0^t e^{As}ds = \int_0^t e^{A(t-s)}ds$. The symbols with hats over them are finite rank infinite dimensional Volterra integral operators defined, over the interval $[0, \tau]$, by $\hat{H}f(t) = \int_0^\tau h(t,s)f(s)ds$ or $\hat{H}f(t) = \int_0^\tau h(s)f(s)ds$, and their kernels are given by: $b_1(s) := e^{A(\tau-s)}B_1$, $c_1(s) := C_1e^{As}$, $d_{12}(s) := C_1\Psi(s)B_2 + D_{12}$ and $d_{11}(t,s) := C_1e^{A(t-s)}\mathbf{1}(t-s)B_1 + D_{11}\delta(t-s)$. Ignoring the dimension of the vectors (matrices) in the domains and ranges, the elements are mapped as follows: $\hat{B}_1 : L^2[0,\tau] \to \mathbb{R}$, $\hat{C}_1 : \mathbb{R} \to \mathbb{R}$, $\hat{D}_{11} : L^2[0,\tau] \to L^2[0,\tau,]$ and $\hat{D}_{12} : \mathbb{R} \to \mathbb{R}$. The above characterization is in terms infinite dimensional, finite rank, operators. In order for the formulation to be usable, the operators must be explicitly evaluated. This was done in [3] for the special case where D_{11} and D_{12} are identically equal to zero and the pairs (A, B_1) and (A, C_1) are controllable and observable respectively.

5 Sampled-Data Problem Formulation

Definition 5.1 *Sampled-data repetitive control design directly addresses continuous-time requirements (including intersample behavior) and results in a controller having a digital repetitive structure.*

Previous works related to sampled-data repetitive control do not satisfy Definition 5.1. In [11] a digital repetitive control problem is considered with a discrete time reference. In [14] the continuous-time requirements are approximated at a sampling rate that is an integer multiple of the actual sampling rate, and no attempt is made to recover a digital repetitive structure. Recall that

$$C(z) = \frac{d(z)}{1 - q(z)z^{-L}}, \tag{7}$$

where $q(z)$ and $d(z)$ are rational functions and the integer L is at least twice the ratio of the sampling frequency, $f_s = 1/\tau$, to the fundamental frequency, f_0, of the the periodic input. The sampling rate, f_s, must be chosen to be an integer multiple of the fundamental frequency, f_0, of the reference/disturbance signal. If this condition is not satisfied, then it is not possible for a controller, with a "low order" digital repetitive structure, to satisfy the continuous-time repetitive performance requirement. Strictly speaking, the order of digital repetitive controllers is *always* at least twice the number of harmonics of interest, due to the term z^{-L}. However, as a practical matter, it is the order of the two rational functions, $q(z)$ and $d(z)$, that determine the computation time (ease of implementation). Indeed, if $q(z) = d(z) \equiv 1$, then only one addition operation is required to implement the controller defined by (7).

5.1 H^∞ Optimal Tracking Problem with Robust Stability

The block diagram for the tracking performance problem with robust stability is shown in Figure 4. The performance weight, W_1, represents the frequency distribution of the signal to be tracked. The robustness weight, W_2, represents the additive uncertainty in the plant P. The sampling pre-filters, F_1 and F_2, are included to ensure that the transfer function from any exogenous input to any sampled output is low pass, which is necessary for the sampling process to be well defined. The H^∞ tracking problem is to minimize the worst case tracking error, z_1, due to two-norm bounded input signals w_1. The robust stability problem is to ensure that the two-norm of the robustness output, $z_2(t)$, satisfies $\|z_2\|_2 < 1$, for all inputs, $w_2(t)$, such that $\|w_2\|_2 < 1$. Thus, $(T_{zw})_{11}$ represents the tracking problem and $(T_{zw})_{22}$ represents the robust stability problem. Thus, solving for $\|T_{zw}\|_\infty < 1$ minimizes the tracking error while guaranteeing robust stability.

Figure 4 has the same form as Figure 1 with the generalized plant, G, given by

$$G_{11} = \begin{bmatrix} W_1 & 1 \\ 0 & 0 \end{bmatrix},\; G_{12} = \begin{bmatrix} P \\ W_2 \end{bmatrix},\; G_{21} = \begin{bmatrix} W_1 F_1 & 0 \\ 0 & F_2 \end{bmatrix},\; G_{22} = \begin{bmatrix} 0 \\ F_2 P \end{bmatrix}.$$

The continuous-time performance/robustness requirements are characterized by the closed loop transfer function T_{zw}. For the problem defined by Figure 4, we have

$$T_{zw} = \frac{1}{1 - P\mathcal{H}C_2\mathcal{S}F_2} \begin{bmatrix} W_1(1 - P\mathcal{H}(C_2\mathcal{S}F_2 - C_1\mathcal{S}F_1)) & 1 \\ W_1W_2(\mathcal{H}C_1\mathcal{S}F_1) & W_2(\mathcal{H}C_2\mathcal{S}F_2) \end{bmatrix}. \tag{8}$$

6 Discrete-Time Equivalent Problem

For (6) to be useful in design, explicit formulas for the associated matrices must be derived in terms of the continuous-time system and performance weights. This was done in [3] for the special case where D_{11} and D_{12} are identically equal to zero and the pairs (A, B_1) and (A, C_1) are controllable and observable respectively. None of these assumptions hold for our sampled-data repetitive control formulation, defined by Figure 4. The characterization of the discrete-time equivalent system given in [13] is completely general, however it is computationally more complicated. Therefore, in this section we directly extend the results of [3] to the general case.

6.1 Explicit formulas for the general case

In this section we give explicit formulas for constructing the discrete-time equivalent system from the continuous-time problem data for the general case. The two operators $[I - \hat{D}_{11}\hat{D}_{11}^*]^{-1}$ and $[I - \hat{D}_{11}^*\hat{D}_{11}]^{-1}$ play very prominent roles in these formulas. The kernels of these two operators are derived in this section. The derivations are analogous to, but considerably more involved than, those in [3] for the special case discussed above. The kernels of these operators are most easily derived in the time domain using differential equation representations. Let $h(t) := \hat{D}_{11}u(t),\ 0 \le t \le \tau$, then

$$\dot{x}_1(t) = Ax_1(t) + B_1u(t),\ h(t) = C_1x_1(t) + D_{11}u(t),\ x_1(0) = 0. \tag{9}$$

The adjoint operator, $\hat{D}_{11}^*,\ 0 \le t \le \tau$, has the differential equation representation

$$\dot{x}_2(t) = -A^*x_2(t) - C_1^*h(t),\ f(t) = B_1^*x_2(t) + D_{11}^*h(t),\ x_2(\tau) = 0, \tag{10}$$

where $f(t) := \hat{D}_{11}^* h(t)$. Thus, $y(t) := [I - \hat{D}_{11}^* \hat{D}_{11}] u(t),\ 0 \le t \le \tau$, is given by

$$\begin{bmatrix} \dot{x}_2(t) \\ \dot{x}_1(t) \end{bmatrix} = \begin{bmatrix} -A^* & -C_1^* C_1 \\ 0 & A \end{bmatrix} \begin{bmatrix} x_2(t) \\ x_1(t) \end{bmatrix} + \begin{bmatrix} C_1^* D_{11} \\ -B_1 \end{bmatrix} u(t), \ \begin{bmatrix} x_2(\tau) \\ x_1(0) \end{bmatrix} = 0,$$

$$y(t) = \begin{bmatrix} B_1^* & D_{11}^* C_1 \end{bmatrix} x(t) + (I - D_{11}^* D_{11})\, u(t), \ \ x(t) := \begin{bmatrix} x_2(t) \\ x_1(t) \end{bmatrix}. \tag{11}$$

The inverse operator, $\hat{R} := [I - \hat{D}_{11}^* \hat{D}_{11}]^{-1}$, exists whenever $\|\hat{D}_{11}\|_\infty < 1$, which is a necessary condition for the existence of a solution to the H^∞ control problem. When the inverse operator, $\hat{R}$, exists it is the solution to the differential equation

$$\dot{x}(t) = A_R x(t) + B_R y(t),\ u(t) = C_R x(t) + D_R y(t),\ x_2(\tau) = 0,\ x_1(0) = 0, \text{ where}$$

$$A_R := \begin{bmatrix} -(A + B_1 D_R D_{11}^* C_1)^* & -(C_1^* C_1 + C_1^* D_{11} D_R D_{11}^* C_1) \\ B_1 D_R B_1^* & (A + B_1 D_R D_{11}^* C_1) \end{bmatrix}, B_R := \begin{bmatrix} C_1^* D_{11} \\ -B_1 \end{bmatrix} D_R,$$

$$C_R := -D_R \begin{bmatrix} B_1^* & D_{11}^* C_1 \end{bmatrix}, \text{ and } D_R := [I - D_{11}^* D_{11}]^{-1}.$$

Let $\Gamma(t) := e^{A_R t}$. If $\Gamma_{11}(\tau)$ is invertible the solution is unique, otherwise any pseudo inverse will give a valid solution. We use $x(t) = e^{A_R(t-t_0)} x(t_0) + \int_{t_0}^t e^{A_R(t-s)} B_R y(s) ds$ to solve for the input, $u(t)$, as a function of the output, $y(t)$. Specifically, we get

$$u(t) = C_R \left\{ -e^{A_R t} \begin{bmatrix} \Gamma_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix} \int_0^\tau e^{A_R(\tau - s)} B_R y(s) ds + \int_0^t e^{A_R(t-s)} B_R y(s) ds \right\} + D_R y(t).$$

So, by inspection, we have that the kernel $r(t,s)$, of the operator $\hat{R}$, is given by

$$r(t,s) = C_R \left\{ -e^{A_R t} \begin{bmatrix} \Gamma_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix} e^{A_R(\tau - s)} B_R + e^{A_R(t-s)} B_R \mathbf{1}(t-s) \right\} + D_R \delta(t-s).$$

The derivation of the other inverse operator, $\hat{Q} := [I - \hat{D}_{11} \hat{D}_{11}^*]^{-1}$, is analogous and only the pertinent definitions will be given. The action of the operator $\hat{Q}$ on a function $f(t)$ is given by $\hat{Q} f(t) = \int_0^\tau q(t,s) f(s) ds$, where the kernel $q(t,s)$ is given by

$$q(t,s) = C_Q \left\{ -e^{A_Q t} \begin{bmatrix} \Upsilon_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix} e^{A_Q(\tau - s)} B_Q + e^{A_Q(t-s)} B_Q \mathbf{1}(t-s) \right\} + D_Q \delta(t-s),$$

$$A_Q := \begin{bmatrix} -(A + B_1 D_{11}^* D_Q C_1)^* & -C_1^* D_Q C_1 \\ (B_1 B_1^* + B_1 D_{11}^* D_Q D_{11} B_1^*) & (A + B_1 D_{11}^* D_Q C_1) \end{bmatrix}, B_Q := \begin{bmatrix} C_1^* \\ -B_1 D_{11}^* \end{bmatrix} D_Q,$$

$$C_Q := -D_Q \begin{bmatrix} D_{11} B_1^* & C_1 \end{bmatrix},\ D_Q := [I - D_{11} D_{11}^*]^{-1}, \text{ and } \Upsilon(t) := e^{A_Q t}.$$

Once again, $q(t,s)$ is a valid representation for $\hat{Q}$ if $\|\hat{D}_{11}\|_\infty < 1$, and it is unique if $\Upsilon_{11}(\tau)$ is invertible. If $\Upsilon_{11}(\tau)$ is singular any pseudo inverse will yield a valid $q(t,s)$.

Theorem 6.1 (discrete-time equivalent, general case) *If* $\|\hat{D}_{11}\|_\infty < 1$ *(a necessary condition for solvability), then the finite dimensional state space representation for the discrete-time equivalent problem (6) can be calculated directly from:* $\acute{A} = \Upsilon_{22}(\tau) - \Upsilon_{21}(\tau)\Upsilon_{11}^{-1}(\tau)\Upsilon_{12}(\tau)$, $\begin{bmatrix} \acute{C}_1 & \acute{D}_{12} \end{bmatrix} = \begin{bmatrix} \Sigma_{CD}^{1/2} & 0 \end{bmatrix} T_{CD}$, $\acute{C}_2 = C_2$

$$\acute{B}_1 = T_B^* \begin{bmatrix} \Sigma_B^{1/2} \\ 0 \end{bmatrix}, \ \textit{and} \ \acute{B}_2 = \begin{bmatrix} (\Phi_{21}(\tau) - \Upsilon_{21}(\tau)\Upsilon_{11}^{-1}(\tau)\Phi_{11}(\tau)) \\ (\Phi_{22}(\tau) - \Upsilon_{21}(\tau)\Upsilon_{11}^{-1}(\tau)\Phi_{12}(\tau)) \end{bmatrix}^T \mathcal{B}_Q - \Psi(\tau) B_2;$$

$$T_B^* \begin{bmatrix} \Sigma_B & 0 \\ 0 & 0 \end{bmatrix} T_B = \Gamma_{21}(\tau)\Gamma_{11}^{-1}(\tau), \quad T_{CD}^* \begin{bmatrix} \Sigma_{CD} & 0 \\ 0 & 0 \end{bmatrix} T_{CD} = \begin{bmatrix} \mathcal{C}_{11} & \mathcal{C}_{12} \\ \mathcal{C}_{12}^* & \mathcal{C}_{22} \end{bmatrix},$$

$$\mathcal{B}_Q^* =: \begin{bmatrix} \mathcal{B}_Q^R & -\mathcal{B}_Q^L \end{bmatrix}, \quad \mathcal{C}_{11} := -\Upsilon_{11}^{-1}(\tau)\Upsilon_{12}(\tau), \quad \mathcal{C}_{12} := -\Upsilon_{11}^{-1}(\tau) \begin{bmatrix} \Phi_{11}(\tau) & \Phi_{12}(\tau) \end{bmatrix} \mathcal{B}_Q,$$

$$\mathcal{C}_{22} := \begin{bmatrix} \mathcal{B}_Q^L & \mathcal{B}_Q^R \end{bmatrix} \left\{ \Phi(\tau) \begin{bmatrix} \Upsilon_{11}^{-1}(\tau) & 0 \\ 0 & 0 \end{bmatrix} \Phi(\tau) - \Omega(\tau) \right\} \mathcal{B}_Q + \tau D_{12}^* D_Q D_{12},$$

$$\Phi(t) := \int_0^t \Upsilon(s) ds, \quad \Psi(t) := \int_0^t e^{As} ds, \quad \Omega(t) := \int_0^t \Phi(r) dr, \quad \mathcal{B}_Q := \begin{bmatrix} -C_1^* D_Q D_{12} \\ B_2 + B_1 D_{11}^* D_Q D_{12} \end{bmatrix},$$

The full detailed proof appears in [17].

If A is well conditioned, $\Psi(\tau)$ can be computed directly from $\Psi(\tau) = A^{-1}(e^{A\tau} - I)$. Similarly, if A_Q is well conditioned, then $\Phi(\tau)$ and $\Omega(\tau)$ can be computed directly from $\Phi(\tau) = A_Q^{-1}(e^{A_Q\tau} - I)$ and $\Omega(\tau) = A_Q^{-1}(\Phi(\tau) - \tau I)$. Numerical evaluation of matrix exponentials is inherently approximate, and the approximation error may be quite significant, see [15]. All the rest of the calculations are simple linear algebra, except for the factorizations required to calculate $\acute{B}_1$, $\acute{C}_1$, and $\acute{D}_{12}$. However, subject to the associated potential numerical difficulties, $\acute{B}_1$, $\acute{C}_1$, and $\acute{D}_{12}$ can be calculated using the singular value decompositions as indicated by the formulas in Theorem 6.1. If $\acute{B}_1$ is a vector, then it can be calculated exactly by taking the square roots of the diagonal elements of $\Gamma_{21}(\tau)\Gamma_{11}^{-1}(\tau)$. Similarly, if $[\acute{C}_1 \;\; \acute{D}_{12}]$ is a vector, then it can also be calculated exactly. The resulting equivalent discrete-time system is the same order as the original continuous-time system, which is a property shared by the more complex characterization of [13].

We have not as yet been able to successfully complete a numerical example verifying that the discrete-time equivalent system of Theorem 6.1 is indeed equivalent with respect to the infinity-norm, even for the simple case of [3]. The use of the noncausal inverse lifting operator, W_τ^{-1}, eliminates from the formulae, for the discrete-time equivalent, all of the observability and controllability grammians normally associated with problems involving delays. This is a very nice feature of the formulation of [3], however it does make one wonder whether we have simplified things to an unrealistic extent. It is precisely the presence of grammian type integrals, involving products of matrix exponentials, that make the discrete-time equivalent of Kabamba and Hara [13] so very difficult to compute. With respect to the sampled-data repetitive control problem, it is irrelevant *how* the discrete-time equivalent system is computed. The solution of the sampled-data repetitive control problem formulated in this paper does not depend on how these issues with respect to computation of discrete-time equivalents are ultimately resolved.

7 On the Discrete-Time Equivalent Problem

Regardless of what method is used to obtain the discrete-time equivalent system, the resulting discrete-time equivalent problem will be the "singular problem," see e.g. [22]. This is an inherent property of well posed sampled-data problems, due to the requirement for all sampled signals to be low pass. The significance of the singular problem is that it cannot be solved by the nice state space methods of [5] for the continuous-time case or [12, 21] for the discrete-time case. There have been several attempts to extend these results to the singular case. None of these methods has evolved to point where their solutions are readily implementable, especially for the discrete-time case. A direct method for H^∞ problems is the use of the bilinear transform. The bilinear transform of a singular problem is, generically, a nonsingular problem. Thus, the bilinear transformation of the discrete-time equivalent problem yields an equivalent continuous time problem, which satisfies the standard assumptions [5]. The equivalence is with respect to the infinity norm [8], thus providing complete equivalence in this case.

8 Continuous-time Equivalent System

Specifically, we convert the discrete-time equivalent problem, $\acute{G}$, to a continuous-time equivalent problem, $G^s = (A^s, B^s, C^s, D^s)$, solve for the corresponding continuous-time controller, $K^s = (A^s_K, B^s_K, C^s_K, D^s_K)$, and convert the controller to its discrete-time equivalent, $K = (A_K, B_K, C_K, D_K)$. The formulas are given by: $A^s := \alpha^{-1}(I + \acute{A})^{-1}(\acute{A} - I)$, $B^s := (I + \acute{A})^{-1}\acute{B}\sqrt{2/\alpha}$, $C^s := \acute{C}(I + \acute{A})^{-1}\sqrt{2/\alpha}$, and $D^s := \acute{D} - \acute{C}(I + \acute{A})^{-1}\acute{B}$; and $A_K := (I + \alpha A^s_K)(I - \alpha A^s_K)^{-1}$, $B_K := (I + A_K)B^s_K\sqrt{\alpha/2}$, $C_K := \sqrt{2}C^s_K(I + A_K)\sqrt{\alpha/2}$, and $D_K := D^s_K + C_K(I + A_K)^{-1}B_K$; where $\alpha > 0$ is the bilinear mapping parameter. It is easily verified that the above bilinear transformations are inverses of each other. The first one maps from the z-domain to the s-domain via $s = \alpha^{-1}(z - 1)/(z + 1)$, such that the unit disk is mapped to the left half plane, and the second one maps from the s-domain to the z-domain via $z = (1 + \alpha s)/(1 - \alpha s)$. The resulting discrete-time controller, $K = (A_K, B_K, C_K, D_K)$, is precisely the controller, $C = [C_1,\ C_2]$, for the sampled-data problem of Figure 4.

9 Recovering a Repetitive Controller Structure

The solution provided by MATLAB's μ-tools toolbox is the so called "central controller", K^c, and all controllers satisfying $\|\mathcal{F}(G^s, K^s)\|_\infty < \gamma$ are parameterized by

$$K^s = \mathcal{F}(K^s, M) := K^c_{11} + K^c_{12}M(I - K^c_{22}M)^{-1}K^c_{21}, \quad \text{where} \|M\|_\infty < \gamma. \tag{12}$$

Whenever W_1 represents a continuous-time repetitive performance requirement, $C = [C_1,\ C_2]$ will be "close" to repetitive, for properly chosen sampling rates. Furthermore, it is likely that, for appropriately selected weighting functions, there will exist a minimal order (same order as the central controller) solution that has a digital repetitive structure. A large class of minimal order controllers is parameterized by all constant matrices M, such that $\|M\|_\infty < \gamma$. Since all of the controllers parameterized by (12) satisfy the robustness and tracking requirements of the H^∞ optimal formulation, we are only concerned with how repetitive (easy to implement) the controllers are. Since there are several free parameters, and the problem is highly nonconvex, traditional search algorithms are not suitable. However, since all solutions are admissible (satisfy the continuous-time requirement imposed by the H^∞ formulation), the new search technique known as genetic algorithms is ideally suited to the task.

9.1 Genetic algorithms

Genetic algorithms (GA's) are so named because they are based on concepts arising in natural selection, see [9, 20] and references therein. GA's have proven to be very effective at finding global, or "near global", maxima for highly nonconvex optimization problems. The basic concept is as follows: (1) Identify the set of free variables (design parameters), which the GA will manipulate as it searches for maxima. (2) Choose an appropriate *fitness* measure (objective function) such that the fitness value is larger for "better" sets of design parameters. (3) Generate an initial *population* of suitable (admissible) sets of design parameters. (4) Apply selection (based on fitness) in conjunction with the "crossover" and "mutation" operations, to the current population, to construct a new population of admissible sets of parameters (the next generation). Step (4) is repeated until "satisfactory" results are obtained. If the fitness measure, crossover and mutation probabilities, and population size are all appropriately chosen, then the algorithm will likely converge to a (hopefully global) maximum. In our case we don't necessarily need to find the global maximum, or even converge; we just want to obtain controllers that are "more" repetitive. Obviously, if a generated "candidate" set of parameters for the next generation is not admissible, i.e. $\|M\|_\infty \geq \gamma$, then it must be either discarded or adjusted (rescaled).

9.2 Repetitive structure via GA's

We propose to use a GA to find controllers K^s such that the bilinear transform, K, of K^s has a digital repetitive structure. The design parameters are the entries of the constant matrix M. A set of parameters (matrix entries) is admissible if $\|M\|_\infty < \gamma$. Let $d(z) =: N_d/D_d$ and $q(z) =: N_q/D_q$, where N_d, D_d, N_q, and D_q are polynomials in z^{-1}. Now, we can write

$$K(z) = \frac{d(z)}{1 - q(z)z^{-L}} = \frac{N_d D_q}{D_d D_q - N_q D_d z^{-L}} =: \frac{N_K}{D_K}. \tag{13}$$

Let the orders of the polynomials be given by $m := ord(D_d D_q)$, $p := ord(N_q D_d)$ and $n := ord(D_K)$. Define the coefficients a_i by $D_K = a_0 + a_1 z^{-1} + \cdots + a_n z^{-n}$, where typically $n = p + L$. If $K(z)$ has a repetitive structure, then there will be a band of zero coefficients, i.e. $a_i = 0, \quad m+1 < i < L-p-1$. Let the normalized coefficients of interest be given by $\hat{a}_j := a_i/a_n, \quad j := i - m - 1, \quad m+1 < i < L-p-1$. Define the *degree of repetitiveness* by $r := L - m - p - 2$. For a given r, let the fitness function be given by

$$f_r(M, \beta) := \left(\beta + \sum_{j=1}^{r} |\hat{a}_j|\right)^{-1}, \tag{14}$$

where the parameter $\beta > 0$ determines the value, β^{-1}, at which the fitness function saturates. Ideally, the GA utilizing the fitness function f_r would seek to maximize the degree of repetitiveness, r. Choosing β to be inversely proportional to r would seem a logical choice.

10 Summary

In this paper we proposed that sampled-data repetitive control design be defined to be design that directly satisfies continuous-time requirements and results in digital repetitive controllers. We posed a two degree of freedom H^∞ optimal sampled-data tracking performance with robust stability problem, see Figure 4. We extended the results of [3], on discrete-time equivalents, to the general case required to solve our H^∞ problem formulation. We also noted that numerical computation of discrete-time equivalents has not yet been resolved for either the approach of [3] or the approach of [13]. Whatever method eventually proves to be practically applicable to computing discrete-time equivalents, the resulting equivalent problem will be singular. We gave explicit state-space formulas for converting the resulting singular discrete-time equivalent problem to an equivalent continuous-time problem satisfying the standard assumptions. There is no loss in design insight due to this conversion, since the actual problem has already been posed in the sampled-data setting. The resulting standard continuous-time equivalent problem can be solved using MATLAB's μ-tools toolbox [1]. The resulting parameterization of all solutions in terms of the central controller can then be exploited to recover a digital controller having a repetitive structure. The corresponding digital repetitive controller can be calculated from the resulting continuous-time controller using the state-space formulas. We posed the problem of searching the parameterization of all solutions for the "most repetitive" solution in terms of genetic algorithms. Numerical investigations await the resolution of the problems with calculating discrete-time equivalent systems.

Acknowledgments

We thank Professor Bamieh and his student Dan Stanek for providing additional details on the proofs in [3] and for valuable feedback on early drafts, respectively. Finally, we thank Mr. LaMoyne Porter for helpful discussions on genetic algorithms.

References

[1] Balas, G., J. Doyle, K. Glover, A. Packard and R. Smith, *μ-Analysis and Synthesis Toolbox: User's Guide.* The Mathworks Inc., Natick, MA, 1991.

[2] Bamieh, B., J. B. Pearson, B. Francis and A. Tannenbaum, A lifting technique for linear periodic systems with applications to sample-data control. *Systems & Control Letters*, 17:79–88, 1991.

[3] Bassam A. Bamieh and J. Boyd Pearson Jr., A general framework for linear periodic systems with applications to H^∞ sampled-data control. *IEEE Transactions on Automatic Control*, 37(4):418–435, 1992.

[4] Chen, T., and B. Francis, Sample-data optimal design and robust stabilization. *ASME Transactions, Journal of Dynamic Systems, Measurement, and Control*, 114:538–543, 1992.

[5] Doyle, J., K. Glover, P. Khargonekar, and B. Francis, State-space solutions to standard H_2 and H^∞ control problems, *IEEE Transactions on Automatic Control*, 34(8):831–847, 1989.

[6] Dullerud, G., and K. Glover, Robust stabilization of sampled-data systems to structured LTI perturbations. *IEEE Transactions on Automatic Control*, 38(10):1497–1508, 1993.

[7] Francis, B., and T. Georgiou, Stability theory for linear time-invariant plants with periodic digital controllers. *IEEE Transactions on Automatic Control*, 33(9):820–832, 1988.

[8] Glover, K., All optimal hankel-norm approximations of linear multivariable systems and their l^∞-error bounds. *International Journal of Control*, 39(6):1115–1193, 1984.

[9] Goldberg, D., *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, New York, NY, 1989.

[10] Hara, S., Y. Yamamoto, T. Omata, and M. Nakano, Repetitive control system: a new type servo system for periodic exogenous signals. *IEEE Transactions on Automatic Control*, 33:659–667, 1988.

[11] Hara, S., H. Kawamura, and H-K. Sung, High-precision tracking by digital repetitive control scheme with periodic-type hold function. In *Proceedings of the American Control Conference*, pages 2676–2680, Chicago, Illinois, 1992.

[12] Iglesias, P., and K. Glover, State-space approach to discrete-time H^∞ control. *International Journal of Control*, 54(5):1031–1073, 1991.

[13] Kabamba, P., and S. Hara, Worst-case analysis and design of sampled-data control systems. *IEEE Transactions on Automatic Control*, 38(9):1337–1357, 1993.

[14] Langari, A., and B. Francis, Sampled-data repetitive control systems. In *Proceedings of the American Control Conference*, pages 3234–3235, Baltimore, MD, 1994.

[15] Moler, C., and C. Van Loan, Nineteen dubious ways to compute the exponential of a matrix. *SIAM Review*, 20(4):801–836, 1978.

[16] Peery, T., and H. Özbay, On H^∞ optimal repetitive controllers. In *Proceedings of the 32nd IEEE Conference on Decision and Control*, pages 1146–1151, San Antonio, Texas, 1993.

[17] Peery, T., *H^∞ Optimal Repetitive Control: Continuous-time and Sampled-data Formulations.* PhD thesis, The Ohio State University, Columbus, OH, June 1995.

[18] Shaw, F., and K. Srinivasan, Analysis and synthesis of discrete-time repetitive controllers. *ASME Transactions, Journal of Dynamic Systems, Measurement, and Control*, 133:216–222, 1991.

[19] Sivashankar, N., and P. Khargonekar, Robust stability and performance analysis of sampled-data systems. *IEEE Transactions on Automatic Control*, 38(1):58–69, 1993.

[20] Srinivas, N., and L. Patnaik, Genetic algorithms: A survey. *IEEE Computer*, pages 17–26, 1994.

[21] Stoorvogel, A., The discrete-time H^∞ control problem with measurement feedback. *SIAM Journal of Control and Optimization*, 30(1):182–202, 1992.

[22] Stoorvogel, A. *The H^∞ Control Problem: A State Space Approach.* International Series in Systems and Control Engineering. Prentice Hall, New York, NY, 1992.

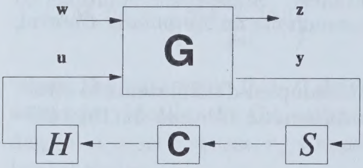


Figure 1: The sampled-data problem.

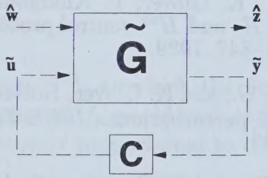


Figure 3: Discrete-time lifted problem.

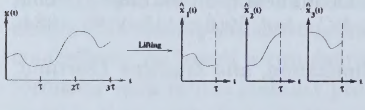


Figure 2: Illustration of the lifting process.

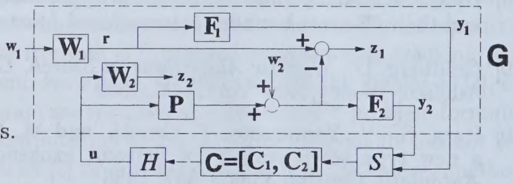


Figure 4: Block diagram of the problem.

Minimum Output Variance Control for FSN Models: Continuous-Time Case

Robert E. Skelton and Guojun Shi
Structural Systems & Control Lab
1293 Potter Engineering Center
Purdue University
West Lafayette, IN 47907

ABSTRACT

In this paper we consider the Finite Signal-to-Noise ratio model for linear stochastic systems. It is assumed that the intensity of noise corrupting a signal is proportional to the variance of the signal. This proportionality constant is called inverse of the signal-to-noise ratio (SNR). Hence, the signal-to-noise ratio of each sensor and actuator is finite - as opposed to the infinite SNR assumed in LQG theory. Computational errors in the controller implementation are treated similarly. The objective is to design a state feedback control law such that the closed loop system is mean square asymptotically stable and the output variance is minimized. The main result is a controller which achieves its maximal accuracy with finite control gains - as opposed to the infinite controls required to achieve maximal accuracy in LQG controllers. Necessary and sufficient conditions for optimality are derived. An optimal control law which involves the positive definite solution of a Riccati-like equation is derived. An algorithm for solving the Riccati-like equation is given and its convergence is guaranteed if a solution exists.

1 Introduction

In the LQG control theory the common model for noise signals is additive noise, where the noise is independent of the signals. This is unrealistic in practice. For instance, a power supply introduces noise that is related to the power required (and hence the "size" of the signal). The same is true for electromechanical devices. In aerospace engineering, spacecrafts are often rotated about their symmetry axis in order to enhance their aerodynamic stability upon reentry. The turbulance subjects the angular rate of the symmetry axis to a turbulent disturbance that depends on some measure of "size" of the precession rates. Thus the assumption that the noise

intensities are independent of the energy of the signal leads to several deficiencies in the practical applications of the LQG theory, because the real sensor and actuator devices have noise sources that are related to the strength of the signal part. In this paper we consider the finite signal-to-noise ratio (FSN) models for linear stochastic systems, which was introduced in [1]. It is assumed that the intensity of the noise is related to the signal covariance in proportion to the noise-to-signal ratio (=SNR^{-1}). The same assumption is made concerning computational errors in controller implementation, since large signals suffer larger computational errors. For detailed descriptions, see [2, 3, 4].

We seek to minimize the variance of the system output y_p. In this optimal control problem, we will demonstrate that these more realistic noise assumptions will keep the control variance relatively small, since a very large control variance comes at the price of very large additive noise, which degrades performance. Hence, in the new theory there is a self-correcting mechanism to keep the control finite in the output variance minimization problem, as opposed to LQG theory, where minimum output variance occurs at infinite control gain. This can have a significant impact on robustness of the controller, since larger control signals have a greater tendency to destabilize unmodeled dynamics in the system, due to "control spillover". Other engineering benefits of the FSN model and the new controller will be discussed in the sequel.

- *The minimum variance is a cost function with physical significance:* Of course, each of the two terms in the cost function of LQG (the output and control variances) have physical significance, but their sum is not a physical entity. Using a non-physical scalar cost function is an artificial "device" used for design. Choosing the LQG weights is a nontrivial task [5, 6]. The new theory treats physically meaningful cost functions (output variance). Weights on the output still must be selected, but weights on control can be zero.

- *The new theory accommodates the singular case:* Setting control penalties to zero in LQG complicates the theory since maximal accuracy occurs at infinite gain (for continuous-time case). Singular solutions are not obtained simply by setting weights to zero in the nominal theory. The new theory accommodates the singular case without modifications.

- *Maximal accuracy occurs with a finite control:* In LQG, the maximal accuracy of the output variance occurs at infinite control effort. Hence some weighting on control in the cost function is necessary for practical implementation. The FSN problem yields a maximal accuracy with finite control effort, hence no weighting on the control signals is required to make the optimization problem well posed.

- *Finite precision controller:* LQG controllers can perform poorly in a finite precision implementation. In fact, the computational errors can dominate the performance [7, 8]. The

FSN controller is optimized with computational errors modeled as white noise with finite signal-to-noise ratios.

All of the above "engineering" improvements over LQG are made possible by the addition of signal dependent noise added to the system model. This is the key difference with LQG. The price we must pay for this improvement is a mild increase in computational complexity. The separation theorem does not hold. The LQG theory is very complete, given the standard modeling assumptions. It is our opinion, however, that many of the practical deficiencies of LQG theory [9] can be traced to the presumption that the signal-to-noise ratio is infinite in the actuators, sensors, and controller computations.

The control problems of FSN models were addressed in [10, 11, 2, 3, 4]. In [2, 4] the covariance control LMI approach was used. In [3] the μ_2 measure was introduced and a $D-K$ iteration approach was used. In this paper, our attention is focused on the state feedback case and the objective is to design an optimal state feedback controller in the sense that the closed loop system is mean square asymptotically (MSA) stable and produces the minimum output variance. Based on a Riccati-like equation, necessary and sufficient conditions for optimality are provided. An algorithm for solving the Riccati-like equation is suggested and the convergence is guaranteed if the FSN system is MSA stabilizable.

The remainder of the paper is organized as follows. The next section introduces the FSN model for linear stochastic systems, and also formulate the minimum output variance control problem. The minimum output variance control problem is solved in terms of a Riccati-like equation in Section 3. Section 4 provides an algorithm for solving the Riccati-like equation and the convergence of the algorithm is discussed. In Section 5, an numerical example is given to illustrate the theory. Finally, conclusions are made in Section 6.

The notation used in this paper is fairly standard. Matrices are denoted by italic letters. For a matrix A, A^T denotes its transpose. The trace of a square matrix S is denoted trS. For symmetric matrices P and Q, $P>0$ $(P\geq 0)$ denotes the fact that P is positive definite (positive semidefinite), $P>Q$ $(P\geq Q)$ denotes $P-Q>0$ $(P-Q\geq 0)$. The notation $||P||$ denotes the spectral norm of the matrix P. For a vector x, $||x||$ denotes the usual Eclidean norm of x. $\mathcal{E}\{\cdot\}$ denotes the expectation operator and $\mathcal{E}_\infty\{\cdot\}$ denotes $\lim_{t\to\infty}\mathcal{E}\{\cdot\}$. $vec\{\cdot\}$ denotes the column-stacking operator. $\otimes$ denotes the Kronecker product.

2 FSN Model and Problem Formulation

CONSIDER the block diagram of a linear system depicted Fig. 1. This figure describes the structure of our model, which includes six zero-mean white noise sources, w_a, w_p, w_s, w_z, w_u, and w_c. Three sources w_a, w_p, w_s are associated with the plant and it's actuating-sensing hardware.

Three sources w_z, w_c, and w_u are computational errors associated with controller implementation. In a digital implementation, w_z would be quantization error in the A/D converter, w_c would be the roundoff errors in the controller gain multiplication, w_u would be the quantization error in the D/A converter. In the analog computations of continuous-time systems w_z, w_c, and w_u are noise sources from the operational amplifiers performing the calculations in the analog computer.

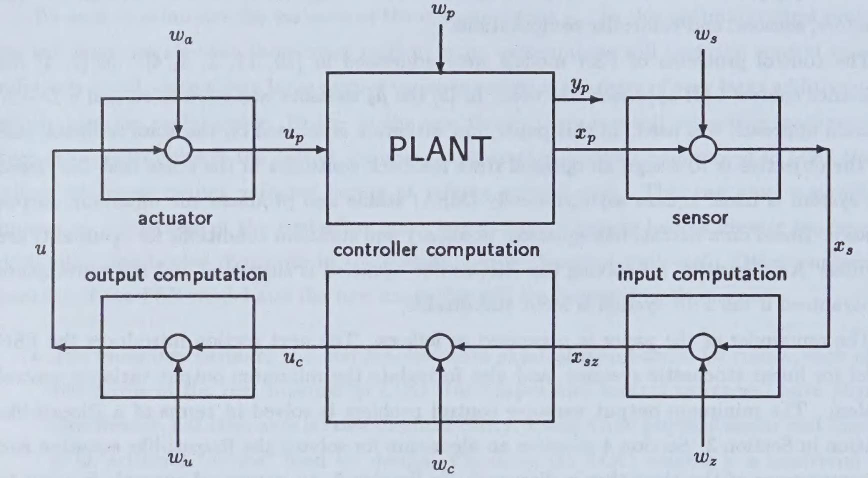


Fig. 1 State Feedback Control for FSN Linear Systems

We assume that the intensity of each noise increases with the size (variance) of the associated uncorrupted signal. Consider the white noise w_a. For our purposes, we let $w_a = w_{a1} + w_{a2}$, where w_{a1} and w_{a2} are independent white noise sources, w_{a1} is the ambient noise with W_{a1}, modeled by the traditional assumptions, and w_{a2} is related to the SNR of the actuator device, defined as follows.

$$\mathcal{E}_\infty\{w_a(t)\} = 0\ ,\quad \mathcal{E}_\infty\{w_a(t)w_a^T(\tau)\} = W_a\delta(t-\tau) = (W_{a1} + W_{a2})\delta(t-\tau)\ , \tag{2.1}$$

where

$$W_{a1} = constant\ ,\quad W_{a2} = \mathcal{C}\{\Sigma_a u(t)\}\ ,\quad \Sigma_a = diag\ \{\cdots\ \sigma_{a_i}\ \cdots\}\ , \tag{2.2}$$

where $\mathcal{C}_\infty\{\cdot\}$ is a special operator denoting covariance of the ***signal part*** of vector $\{\cdot\}$, and $\sigma_{a_i}^{-1}$ is the SNR for the i^{th} actuator. The inverse of the SNR will be called the noise-to-signal ratio (NSR). All noises are uncorrelated with the initial plant state $x_p(0)$, which is zero mean. The noise-to-signal ratio (NSR) of all hardware and computational devices enters into the model by relating the intensity of the noise source to the covariance of the signal as in (2.2). For simplicity, we assume that all of the channels in a given hardware have the same SNR, *e.g.*, $\Sigma_a = \sigma_a I$.

In Fig. 1, all of the signals are related by

$$\begin{aligned}
\dot{x}_p &= A_p x_p + B_p u_p + w_p \, , \\
y_p &= C_p x_p \, , \\
u_p &= u + w_a \, , \\
x_s &= x_p + w_s \, , \\
x_{sz} &= x_s + w_z \, , \\
u &= u_c + w_u \, , \\
u_c &= G x_{sz} + w_c \, ,
\end{aligned} \tag{2.3}$$

where $x_p \in \Re^{n_p}$ is the plant state, $u_p \in \Re^{n_u}$ is the control signal to the plant, $y_p \in \Re^{n_y}$ is the output to be regulated, $x_s \in \Re^{n_p}$ is the measurement of the states, $x_{sz} \in \Re^{n_p}$, $u \in \Re^{n_u}$, $u_c \in \Re^{n_u}$ are the signals indicated in Fig. 1, A_p, B_p, and C_p are real constant matrices with appropriate dimensions, $G \in \Re^{n_u \times n_x}$ is the controller gain to be designed, and the noises

$$\begin{aligned}
w_p &= D_p w_{p1} + w_{p2} \, , \\
w_a &= w_{a1} + w_{a2} \, , \\
w_s &= w_{s1} + w_{s2} \, , \\
w_z &= w_{z1} + w_{z2} \, , \\
w_c &= w_{c1} + w_{c2} \, , \\
w_u &= w_{u1} + w_{u2}
\end{aligned} \tag{2.4}$$

are zero-mean white noises. For $k = p,\ a,\ s,\ z,\ c,\ u$, the intensity of w_{k1}, is $W_{k1} > 0$. It is assumed in the state feedback case of this paper that $w_{s1} = 0$, $w_{z1} = 0$. Now one can obtain the closed loop system from the equations in (2.3)

$$\dot{x}_p = (A_p + B_p G) x_p + w \, , \tag{2.5}$$

$$y_p = C_p x \, , \tag{2.6}$$

where

$$w \triangleq B_p(w_a + w_u + w_c + G w_s + G w_z) + w_p \, . \tag{2.7}$$

Suppose the state covariance of the closed loop system, $X_p \triangleq \mathcal{E}_\infty\{x(t)x^T(t)\}$, exists. Then using the operator $\mathcal{C}_\infty\{\cdot\}$, one can show that

$$\begin{aligned}
W_{a2} &= \mathcal{C}_\infty\{\sigma_a u(t)\} \\
&= \sigma_a^2 \mathcal{C}_\infty\{G(x_p + w_s + w_z) + w_c + w_u\} \\
&= \sigma_a^2 \mathcal{E}_\infty\{G x_p\} \\
&= \sigma_a^2 G X_p G^T \, .
\end{aligned} \tag{2.8}$$

Similarly,

$$\begin{aligned} W_{u2} &= \mathcal{C}_\infty\{\sigma_u u_c(t)\} = \sigma_u^2 G X_p G^T \ , \\ W_{p2} &= \mathcal{C}_\infty\{\sigma_p x_p(t)\} = \sigma_p^2 X_p \ , \\ W_{s2} &= \mathcal{C}_\infty\{\sigma_s x_p(t)\} = \sigma_s^2 X_p \ , \\ W_{z2} &= \mathcal{C}_\infty\{\sigma_z x_s(t)\} = \sigma_z^2 X_p \ , \\ W_{c2} &= \mathcal{C}_\infty\{\sigma_c G x_{sz}(t)\} = \sigma_c^2 G X_p G^T \ . \end{aligned} \tag{2.9}$$

Hence, the intensity of the white noise w in (2.5) is given by

$$W(X_p,\, G) = \sigma^2 B_p G X_p G^T B_p^T + \sigma_p^2 X_p + B_p W_{auc} B_p^T + D_p W_{p1} D_p^T \tag{2.10}$$

where

$$W_{auc} \triangleq W_{a1} + W_{u1} + W_{c1} \ , \quad \sigma^2 \triangleq \sigma_a^2 + \sigma_c^2 + \sigma_s^2 + \sigma_z^2 + \sigma_u^2 \ .$$

The scalar σ^{-2} is referred to as the INFORMATION QUALITY of the FSN system. Consider the closed-loop system (2.3). It can be shown that the state covariance X_p satisfies the following Lyapunov equation

$$(A_p + B_p G) X_p + X_p (A_p + B_p G)^T + W(X_p,\, G) = 0 \ . \tag{2.11}$$

This matrix equation is not the conventional Lyapunov equation since $W(X_p,\, G)$ is a function of covariance X_p. Substituting for $W(X_p,\, G)$ into the equation (2.11) and collecting terms, we have a Lyapunov-like equation

$$\begin{aligned} \mathcal{L}(X_p,\, G) \ \triangleq \ & (A_o + B_p G) X_p + X_p (A_o + B_p G)^T + \\ & \sigma^2 B_p G X_p G^T B_p^T + B_p W_{auc} B_p^T + D_p W_{p1} D_p^T = 0 \ . \end{aligned} \tag{2.12}$$

where $A_o \triangleq A_p + \frac{1}{2}\sigma_p^2 I_{n_p}$.

We shall make the following assumptions about the FSN model:

Assumption 2.1 *(i)* *The pair $(A_p,\, B_p)$ is controllable;*
(ii) *The pair $(C_p,\, A_p)$ is observable;*
(iii) *The input matrix B_p has full column rank.*

Definition 2.1 *The FSN closed loop system is mean square asymptotically (MSA) stable if its steady state covariance X_p exists and is positive definite.*

Definition 2.2 *The FSN open loop system is mean square asymptotically (MSA) stabilizable if there exists a gain G such that the FSN closed loop system is MSA stable.*

In this paper it is also assumed that a noisy measurement of the state of the FSN system is available for feedback. We shall consider the following problem.

Minimum Output Variance Control Problem:

Consider the FSN system (2.3). Find a MSA stabilizing state feedback controller $u = G(x_p + w_s + w_z) + w_c + w_u$ such that the output variance $tr[C_p X_p C_p^T]$ is minimized subject to (2.12).

3 Minimum Output Variance Controller Design

In this section we shall solve the minimum output variance control problem for the FSN systems and reveal some features of the optimal controller.

Proposition 3.1 *Consider the FSN system and let a controller gain G be given. Then the following statements are equivalent:*

(i) The FSN system is MSA stabilized by G;

(ii) The matrix

$$\mathcal{A} \triangleq I_{n_p} \otimes (A_o + B_pG) + (A_o + B_pG) \otimes I_{n_p} + \sigma^2 (B_pG) \otimes (B_pG) \tag{3.1}$$

is stable;

(iii) The Lyapunov-like equation (2.12) has a solution $X_p > 0$;

(iv) The dual Lyapunov-like equation

$$(A_o + B_pG)^T Y_p + Y_p(A_o + B_pG) + \sigma^2 G^T B_p^T Y_p B_p G + C_p^T C_p = 0\,. \tag{3.2}$$

has a solution $Y_p > 0$.

The following theorem is our main result, which provides the minimum output variance optimal controller for FSN systems.

Theorem 1 *The minimum output variance control problem has a solution if and only if the Riccati-like equation*

$$A_o^T Y_p + Y_p A_o - \sigma^{-2} Y_p B_p (B_p^T Y_p B_p)^{-1} B_p^T Y_p + C_p^T C_p = 0 \tag{3.3}$$

has a solution $Y_p > 0$, where $A_o = A_p + \frac{1}{2}\sigma_p^2 I_{n_p}$. In this case, the FSN optimal controller is given by

$$G = -\sigma^{-2}(B_p^T Y_p B_p)^{-1} B_p^T Y_p\,, \tag{3.4}$$

the minimum output variance by

$$J_y(Y_p) = tr[Y_p(B_p W_{auc} B_p^T + D_p W_{p1} D_p^T)]\,, \tag{3.5}$$

and the control effort is defined by

$$J_u(X_p, G) \triangleq \mathcal{C}_\infty\{u_p^T u_p\} = tr[G X_p G^T]\,, \tag{3.6}$$

where X_p is the solution of the Lyapunov-like equation (2.12).

Remark 3.1 *Suppose the minimum output variance control problem has a solution. Then the nominal system matrix* A_p+B_pG *has a stability margin* $\frac{1}{2}\sigma_p^2$. *That is, all eigenvalues of* A_p+B_pG *lie to the left of the vertical line* $-\frac{1}{2}\sigma_p^2+j\omega$.

Remark 3.2 *Note from (3.4) and (3.5) that only* σ_p^2 *and the information quality* σ^{-2} *influence the optimal control gains. The gain increases with information quality* σ^{-2}.

The Riccati-like equation is not the standard Riccati equation and hence can not be solved directly. In the next section, a successive approximation algorithm will be provided.

4 Successive Approximation Algorithm

As can be seen from Theorem 1, the minimum output control problem involves the positive solution Y_p of the Riccati-like equation (3.3). An iteration algorithm was proposed in [12] for solving this kind of Riccati-like equation with control penalty $R \neq 0$. For the Riccati-like equation with control penalty $R=0$, we shall give a modified algorithm.

Lemma 1 *Suppose that the Riccati-like equation (3.3) has a solution* $Y_p > 0$. *Then, for an arbitrary* $0 < R < \infty$, *the following Riccati-like equation*

$$A_o^T S + SA_o - SB_p(R+\sigma^2 B_p^T S B_p)^{-1} B_p^T S + C_p^T C_p = 0 \tag{4.1}$$

has a solution $S > Y_p > 0$.

Algorithm I

Let $(A_p,\ B_p,\ C_p,\ \sigma^2,\ \sigma_p^2)$ be given. Set index $i=0$ and $S_0=0$. Choose a prescribed tolerance $\epsilon > 0$, matrix $R > 0$, and the maximal iteration number N_{max}. For $i \geq 0$, solve the Riccati equations

$$A_o^T S_{i+1} + S_{i+1}A_o - S_{i+1}B_pR_i^{-1}B_p^T S_{i+1} + C_p^T C_p = 0 \tag{4.2}$$

$$R_i = R + \sigma^2 B_p^T S_i B_p \tag{4.3}$$

to obtain a sequence of positive definite solutions $\{S_i\ :\ i=1,\ 2,\ \ldots\}$ until $\|S_{i+1}-S_i\| < \epsilon$ or $i > N_{max}$.

□

Suppose that $S \triangleq \lim_{i\to\infty} S_i > 0$ exists. Then we propose the following algorithm for solving the Riccati-like equation (3.3).

Algorithm II

Step 0 Solve **Algorithm I** to obtain $S > 0$.

Step 1 Set index $i = 0$ and $Y_p^0 = S$. Choose a prescribed tolerance $\epsilon > 0$ and the maximal iteration number N_{max}. For $i \geq 0$, solve the Riccati equations

$$A_o^T Y_p^{i+1} + Y_p^{i+1} A_o - Y_p^{i+1} B_p (\sigma^2 B_p^T Y_p^i B_p)^{-1} B_p^T Y_p^{i+1} + C_p^T C_p = 0 \tag{4.4}$$

to obtain a sequence of positive definite solutions $\{Y_p^i \, : \, i = 1, \, 2, \, \ldots\}$ untill $\|Y_p^{i+1} - Y_p^i\| < \epsilon$ or $i > N_{max}$.

□

Theorem 2 *The minimum output variance control problem has a solution if and only if Algorithm II converges.*

Conclusions

This paper gives an algorithm to solve the minimum output variance problem using state feedback and FSN models of linear systems.

References

[1] R. E. Skelton. Robust control of aerospace systems. In *IFAC Symposium on Robust Control Design*, pages 24–32, Rio De Janeiro, Brasil, September 1994.

[2] Guojun Shi and Robert E. Skelton. State feedback covariance control for linear finite signal-to-noise ratio models. In *Proc. 34th IEEE Conf. Decis. and Contr.*, New Orleans, Louisiana, December 1995.

[3] Uffe H. Thygesen and Robert E. Skelton. Linear systems with finite signal-to-noise ratios: A robustness approach. In *Proc. 34th IEEE Conf. Decis. and Contr.*, New Orleans, Louisiana, December 1995.

[4] Robert E. Skelton and Guojun Shi. Finite signal-to-noise ratio models: covariance control by state feedback. In *1995 European Contr. Conf.*, Rome, Italy, September 1995.

[5] C. Hsieh, R. E. Skelton, and F. M. Damra. Minimum energy controllers with inequality constraints on output variances. *Optimal Control Applications & Methods*, 10:347–366, 1989.

[6] Guoming Zhu, Mario A. Rotea, and Robert E. Skelton. A convergent feasible algorithm for the output covariance constraint problem. In *1993 American Contr. Conf.*, pages 1675–1679, San Francisco, CA, June 1993.

[7] Michel Gevers and Gang Li. *Parameterizations in Control, Estimation and Filtering Problems.* Springer-Verlag, New York, 1993.

[8] Ketao Liu, Robert E. Skelton, and Karolos Grigoriadis. Optimal controllers for finite wordlength implementation. *IEEE Trans. Automat. Contr.*, AC-37(9):1294–1304, 1992.

[9] R. E. Skelton. Model error concepts in control design. *Int. J. Contr.*, 49(5):1725–1753, 1989.

[10] W. M. Wonham. Optimal stationary control of a linear system with state-dependent noise. *SIAM J. Contr.*, 5(3):486–500, 1967.

[11] D. L. Kleinman. Optimal stationary control of linear systems with control-dependent noise. *IEEE Trans. Automat. Contr.*, AC-14(6):673–677, 1969.

[12] D. L. Kleinman. Numerical solution of the state dependent noise problem. *IEEE Trans. Automat. Contr.*, AC-21(6):419–420, 1976.

Toward Software-Only Low-Delay Video Encoding*

Toby Berger and Yi-Jen Chiu

School of Electrical Engineering
Cornell University
Ithaca, New York 14853
berger/yjchiu@ee.cornell.edu

Abstract

The promise of ubiquitous duplex video communications remains unrealized. We discuss the three-way tradeoff between encoder complexity, data rate and video quality. We have devised a transform-free approach to the encoding of difference frames that compares favorably with H.261. Also, we have a 30 fps algorithm that eschews both motion compensation and transforms at the expense of a modest increase in data rate. This algorithm is a feasible candidate for software-only encoding of 30 fps CIF-sized color video on existing workstations, assuming provision has been made for A/D converted camera output to be pipelined into the workstation in real time.

1 Background

The age of digital video communications is arriving slower than many had anticipated. Picturephone (1970s) and videophone (1990s) were not commercial successes because they did not provide full color, full motion video and were not cost effective. Desktop video, using windows in computer monitors or TV screens, has the advantage that the most expensive component - the display - is already present in tens of millions of homes and offices. However, the current generation of full-motion desktop video systems requires special purpose chips to perform the encoding operations. The chips usually come mounted on video boards that are expensive and

---

*This work was supported in part by the National Science Foundation under grants NCR-9216975 and IRI-9310670

whose installation intimidates most users. Video decompression is 50 to 100 times less computationally intensive than video compression, as a consequence of which satisfactory software-only decoders already exist as freeware on the net.

2 Reduced Computation Video Compression Algorithms

The most computationally intensive operations in digital video encoding are the 2D DCTs and especially the motion estimation. We have found a family of low-delay video encoding algorithms suitable for interactive applications such as videotelephony and videoconferencing that use no transforms. Moreover, they compare favorably with H.261 in the tradeoff between data rate and video quality. We also have studied modifications of these algorithms that employ no motion compensation, thereby reducing computational complexity sufficiently that existing workstations can encode 30 CIF-sized color frames per second in real time using no hardware other than an A/D interface between the camera and the CPU. The price paid for eliminating motion compensation is an increase in data rate by a factor of roughly 1.7 for head-and-shoulders videos, but the data rate is still well below that of frame-by-frame JPEG of comparable quality and the computational burden is drastically reduced. Workstations linked by wideband LANs therefore should be able to video communicate with one another using this software-only algorithm.

3 Dependencies Among Difference Frames

Figure 1 shows difference frames 7 and 8 computed during relatively high bitrate motion compensated compression of the standard Miss America video. During and between these two frames, she is both speaking and blinking her eyes. These two difference frames exhibit strong spatial dependencies and mild intensity dependencies. In this regard they are typical of successive difference frames in any head-and-shoulders motion compensated video sequence. Such dependencies contrast with the nearly independent residuals in well-tuned DPCM coding of speech. Taking advantage of these dependencies cleverly can reduce the data rate needed for video transmission, especially when one considers that the coding of approximations to difference frames usually accounts for more than 80% of the data

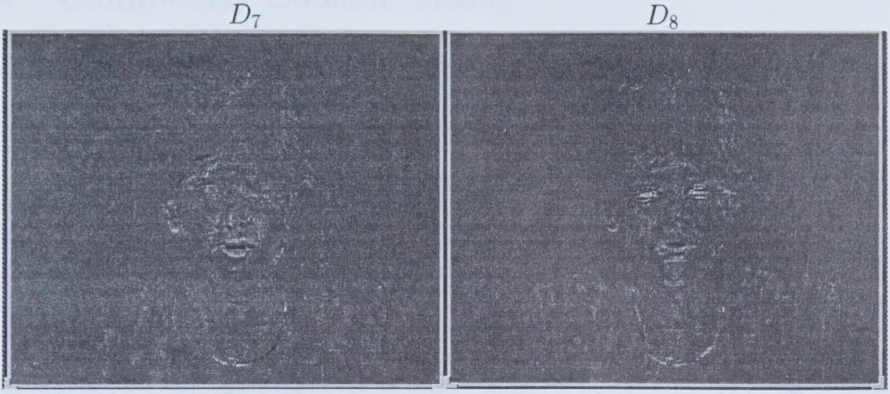


Figure 1: Two Successive Difference Frames of Miss America when Blinking

rate, with the motion vectors usually accounting for most of the remaining 20%. We remark, however, that difference frames cannot be compensated by the block matching algorithms used for motion compensation of the actual frames because corresponding lines on successive difference frames may change intensity. Note, for example, that the dress neckline on the right as side of the frame is dark in difference frame 7 and light in difference frame 8. Similarly, certain lines in the vicinity of the mouth have changed polarity.

Figure 2 repeats difference frame 8 and also shows difference frame 8 for an algorithm that uses direct subtraction without any motion compensation. The uncompensated difference frame contains more energy than does the compensated one and is characterized by "thickening" of certain key lines. Successive such uncompensated difference frames also exhibit strong spatial and mild intensity dependencies similar to those of their motion compensated counterparts.

Figure 3 shows a decomposition of the uncompensated difference frame 8 into what we shall term its translative component and its innovative component. The desire is to make the innovative component consist of those pixels at which something **new** has taken place, such as the deformation of lips, the blinking of eyes or the revelation of background at the trailing edge of a moving object. We have a simple technique for deciding which pixels to include in the innovative component. Although certainly not immune to mistakes, our technique has the advantage of not involving any object identification or scene analysis.

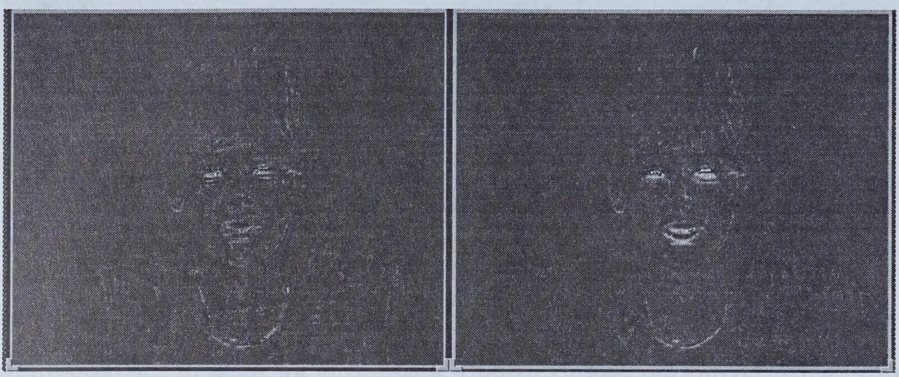
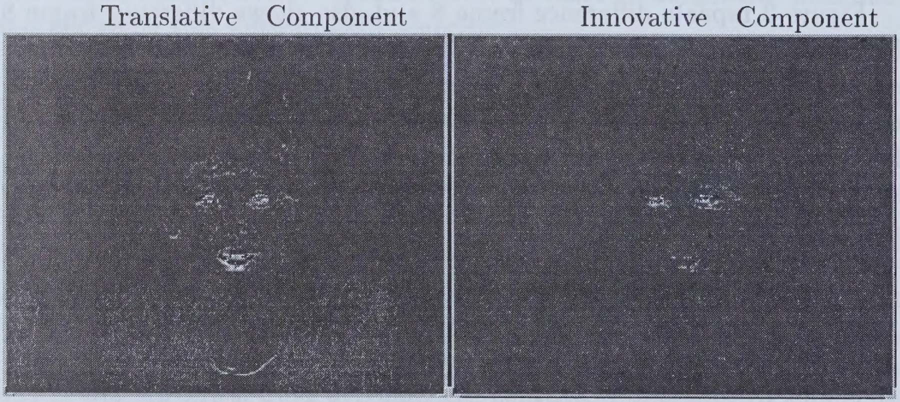


Figure 2: D_8 with and without Motion Compensation



Translative Component
Innovative Component

Figure 3: Decomposition of D_8

4 Complexity Considerations

If one does the DCT in standardized digital video coding in $n \times n$ blocks, then theory suggests that this requires $2\log_2 n$ additions and $2\log_2 n$ multiplications per pixel. For small n, however, including the value $n = 8$ usually used in practice, most of the multiplications can be converted to additions. Accordingly, the actual computational burden of the DCT is better described as about 10 additions per pixel per frame. The per-pixel-per-frame computational requirement of hierarchical motion estimation with h block locations explored for each motion vector calculated is h logical operations (rectifications) and $2h$ additions (a subtraction before the rectification and an accumulation after the rectification). (The number of additions can be reduced if one capitalizes on the overlap of the various blocks visited during the search, but this imposes other bookkeeping requirements and therefore tends not to provide meaningful computational relief.) A typical value of h is 30, so hierarchical motion estimation consumes 60 additions and 30 rectifications per pixel. In contrast, the various instantiations of our direct subtraction algorithm require only one arithmetic and one logical operation per pixel per frame, plus another arithmetic and roughly two more logical operations that need to be performed for fewer than 5% of the pixels.

5 Comparisons with H.261 and JPEG

Video performance comparisons tend to be either subjective or based on inappropriate quantitative measures such as signal-to-noise power ratios. Most of the tests we have performed so far have been in the high bitrate, high quality regime. We process 30 fps of either CIF or NTSC video, sometimes color and sometimes B/W. Most observers consider the reconstructions in our tests obtained with our algorithm, with H.261 and with JPEG in these tests to be close to indistinguishable from the original and hence from each other. PSNR calculations show that the motion-compensated without DCT versions of our algorithm outperform H.261 by approximately 1 db while saving about 20% in data rate and 10% in computational effort.

Our direct subtraction algorithm, on the other hand, offers substantial improvements. Since no motion compensation occurs, it is inappropriate to compare it with H.261 which is substantially more computationally intensive at the encoder. Accordingly, we compare it with JPEG-on-the-

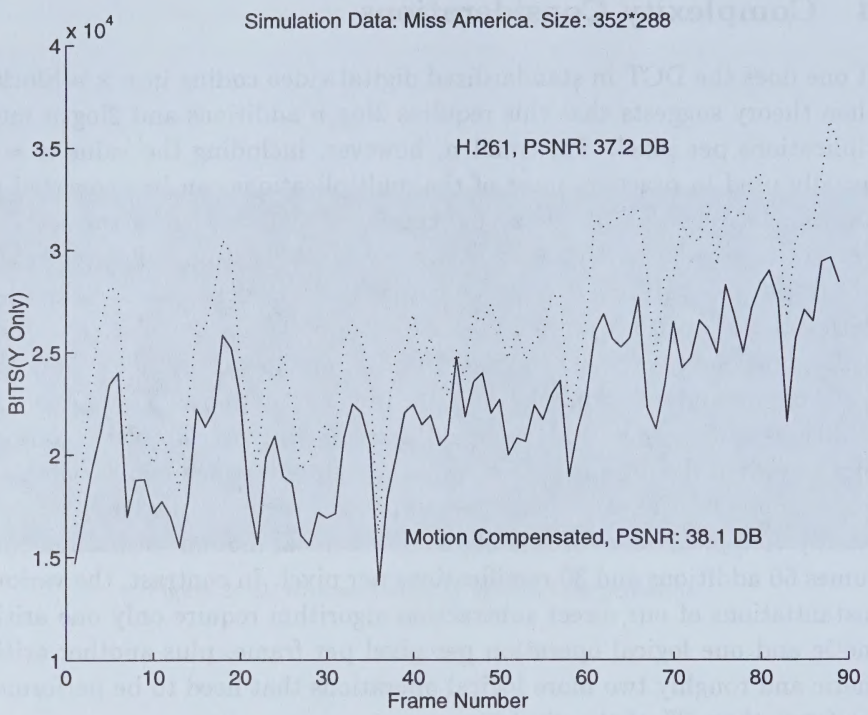


Figure 4: Comparison between H.261 and motion compensated algorithm

fly, which is computationally much less intensive than H.261 at the expense of poorer compression ratio. We find that, when the Q-factor in JPEG and the parameters in our direct subtraction algorithm are adjusted to give the same PSNR, our compressed bitrate is 3 to 7 times lower [1] than that of JPEG. Moreover, the peak bitrate for any frame stays around half of the JPEG bitrate, assuming no abrupt scene changes. (See Figure 5.) As cited in the abstract, these mean bitrates are about 1.5 to 1.7 times greater than that of H.261 for the same high quality. The computational effort, moreover, is cut by nearly a factor of 10 relative to JPEG and by several 10's relative to any motion-compensated scheme.

6 Other Considerations

Complexity, bitrate and video quality tell much but not all of the story. Other important considerations include rate scalability and flow control.

---

[1] This factor varies with the video. We have used the "standard" Miss America, Salesman, Mom, Mom and Granddaughter, and Grandma videos, plus some locally generated video clips in our experiments.

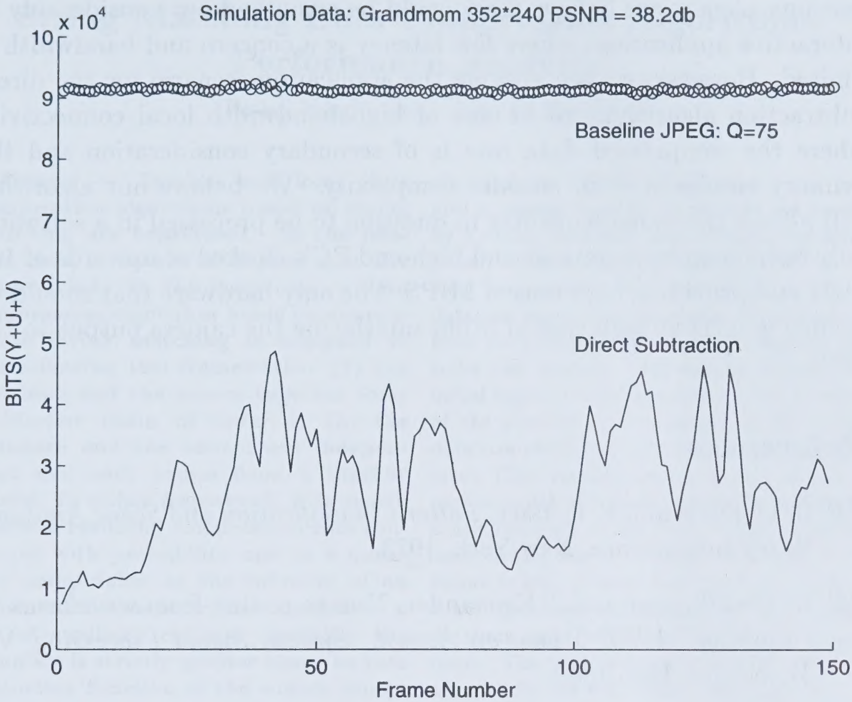


Figure 5: Comparison between JPEG and direct subtraction algorithm

We shall not discuss rate scalability here except to say that both our motion compensated and our direct subtraction algorithms have parameters in them that can be either manually or adaptively set to adjust the mean bitrate.

Figure 4 shows compressed bits vs frame number for the Miss America sequence for H.261 and for one of our motion compensated algorithms when no flow control is applied. They are comparable in sample variance and peak deviation, so we are confident that short-term rate fluctuations in our algorithm could be moderated by buffer flow control techniques similar to those currently being employed in commercial versions of H.261 videoconferencing systems, with a comparable degradation in picture quality.

Figure 5 shows compressed bits vs frame number for the Grandma sequence for two of our direct subtraction algorithms, again in the absence of any flow control. The sample variance and peak deviation are not appreciably larger than those associated with the motion-compensated versions. Again, we believe these could be smoothed out considerably in interactive applications where low latency is a concern and bandwidth is limited. However, we are viewing the application scenario for the direct subtraction algorithms to be one of high bandwidth local connectivity where the compressed data rate is of secondary consideration and the primary emphasis is on encoder complexity. We believe our algorithm will permit the video sequences in question to be processed in a software-only fashion on workstations and high-end PC's clocked at upwards of 100 MHz and providing high tens of MIPS. The only hardware that should be needed is A/D at high tens of Mbps interfacing the camera output to the CPU.

References

[1] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley Interscience, New York, 1973.

[2] R. Forchheimer and T.Kronander, "Image coding-from waveforms to animation",*IEEE Trans. on Acoust., Speech, Signal Processing*, Vol 37, No. 12, Dec. 1989.

[3] A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall 1989.

[4] D. LeGall, "MPEG: a video compression standard for multimedia applications," *Commun. ACM*, vol. 34, no. 4, April 1991.

[5] Haibo Li, Astrid Lundmark, and Robert Forchheimer, "Image sequence coding at very low bitrate: A review," *IEEE Trans. Image Processing* , Vol. 3 No. 5, pp. 589-609, Sep. 1994.

[6] M. Liou, "Overview of the p*64 Kbit/s Video Coding Standard", *Commun. ACM*, vol. 34, no. 4, pp. 59-63, April 1991.

[7] G.K. Wallace, "The JPEG still picture compression standard," Commun. ACM, vol.34, no. 4, pp. 30-44, Apr. 1991.

[8] "Special issue on advances in image and video compression",*IEEE Proc.*, vol. 83, no. 2, pp134-340, Feb., 1995.

[9] "Special issue on digital telvision, Part 1: Enabling technologies",*IEEE Proc.*, vol. 83, no. 6, pp829-1004 June 1995.

String Matching Data Compression Algorithms: Performance Analysis*

En-hui Yang[1] and John C. Kieffer[2]

Abstract — **Lossless and lossy data compression algorithms based on string matching are considered. In the lossless case, a result of Wyner and Ziv is extended. In the lossy case, a data compression algorithm based on approximate string matching is analyzed in the following two frameworks: (1) the database and the source together form a Markov chain of order k; (2) the database and the source are independent and each comes from a Markov model. In either framework, it is shown that the resulting compression rate converges with probability one to a quantity computable as the infimum of an information-theoretic functional over a set of auxiliary random variables; the quantity is strictly greater than the rate distortion function of the source except in some symmetric cases. In particular, this result implies that the lossy algorithm proposed by Steinberg and Gutman is not optimal, even for memoryless or Markov sources.**

I. INTRODUCTION

Starting with the work of Ziv and Lempel, string matching has become a basic concept in the area of data compression. Many data compression algorithms based upon string matching have been proposed and studied [10] [6] [8] [5] [3] [11] [12]. The manner in which string matching is used for compression varies from algorithm to algorithm. However, one can give a general formulation of what one means by a string matching data compression algorithm, whether one is interested in lossless or lossy compression. One starts with an initial database consisting of a sequence of symbols from the source alphabet, known to both encoder and decoder. The encoder encodes an initial segment of the source sequence in terms of the position of a substring of the initial database providing a good match to this segment. (The matching substring will provide a perfect match in lossless coding, and will lie in a neighborhood of the source segment in the lossy coding case.) There is an update rule, known to both encoder and decoder, via which a new database is computed from the initial database and the encoded initial source segment. The new database is used to provide a match for the next source segment, and the encoding proceeds.

We formalize the general notion of string-matching data compression algorithm discussed in the preceding paragraph. Let x be a source sequence to be compressed by a string-matching data compression algorithm. A sequence of triples $(x^{(i)}, y^{(i)}, b^{(i)}), i = 1, 2, \ldots$ is then produced in which:

(a.1) Each $x^{(i)}$ is a sequence of finite length. The sequences $\{x^{(1)}, x^{(2)}, \ldots\}$ are the source segments sequentially processed by the encoder; these sequences, in the order indicated, form a partition of x.

(a.2) $\{y^{(1)}, y^{(2)}, \ldots\}$ are the databases employed. Each $y^{(i)}$ is a finite or infinite sequence of symbols from the source alphabet. The initial database $y^{(1)}$ is known to both encoder and decoder.

(a.3) For each i, $b^{(i)}$ is the binary codeword which is the encoder output in response to encoder input $x^{(i)}$. The string $b^{(i)}$ is determined by the position in $y^{(i)}$ of a string matching $x^{(i)}$. (If there is no such

---

*This work was supported in part by NSF under Grants NCR-9304984, NCR-9205265, and NCR-9508282.

[1] En-hui Yang is with Dept. of Mathematics, Nankai University, Tianjin 300071, P. R. China. Email: ehyang@ee.umn.edu

[2] John C. Kieffer is with Dept. of Electrical Engineering, University of Minnesota, 200 Union Street SE, Minneapolis, MN 55455, USA. Email: kieffer@ee.umn.edu

string, $b^{(i)}$ is determined by applying a "default encoder" to $x^{(i)}$.) The pair $(b^{(i)}, y^{(i)})$ determines $y^{(i+1)}$ by a mechanism known to both encoder and decoder.

We present three examples of lossless string-matching data compression algorithms. Although they are quite different, each of them can be formulated according to the general formulation given previously.

Example 1. In this first example, studied in [11] [12], the size of the database is a fixed positive integer n. The source sequence is an infinite sequence $x = (x_0, x_1, \ldots)$. We initialize by taking the initial database to be a sequence $y^{(1)} = (x_{-n}, x_{-n+1}, \ldots, x_{-1})$, and by taking the initial source segment to be $x^{(1)} = x_0$. Suppose $x^{(i)} = (x_j, \ldots, x_k)$. Then $y^{(i+1)} = (x_{k-n+1}, \ldots, x_k)$ and $x^{(i+1)}$ is the longest initial segment of $(x_{k+1}, x_{k+2}, \ldots)$ which is a substring of $y^{(i+1)}$ or else is equal to x_{k+1} if there is no such substring. If $k - j > 1$, then $x^{(i)}$ is a substring of $y^{(i)}$, and the codeword $b^{(i)}$ is obtained by encoding the pair $(k - j, m)$, where m is the smallest positive integer such that $(x_{j-m}, \ldots, x_{k-m}) = x^{(i)}$; otherwise, $b^{(i)}$ is obtained by encoding the pair $(1, x_k)$. The performance of this algorithm has been examined for the case in which the database samples and the source samples are drawn from a common memoryless model. In this case, the compression rate in code bits per source sample, averaged over the entire source sequence, is almost surely equal to a constant C_n. As $n \to \infty$, the compression rate C_n converges to the entropy rate of the underlying memoryless model.

Example 2. In this example, the size of the database $y^{(i)}$ grows as $i \to \infty$. Let n, m be positive integers. The source sequence is an infinite sequence $x = (x_0, x_1, \ldots)$. The initial database is a sequence $y^{(1)} = (x_{-n}, x_{-n+1}, \ldots, x_{-1})$. The source segments $\{x^{(i)} : i = 1, 2, \ldots\}$ are the blocks of length m which partition the source sequence. Suppose $x^{(i)} = (x_j, \ldots, x_k)$. Then $y^{(i)} = (x_{-n}, \ldots, x_{j-1})$ and $y^{(i+1)} = (x_{-n}, \ldots, x_k)$. In other words, the current source segment is appended to the current database to obtain the next database. The codewords $\{b^{(i)}\}$ are obtained much as in Example 1, except that the decoder doesn't have to be furnished with the length of each $x^{(i)}$. As with Example 1, the performance in Example 2 has been examined for the case in which the database samples and the source samples are drawn from a common memoryless model. In this case, the average compression rate is almost surely equal to a constant D_m. (The constant D_m turns out not to depend upon the size n of the initial database.) The compression rate D_m tends as $m \to \infty$ to the entropy rate of the underlying memoryless model.

Example 3. In this example, we use an infinite database to encode a source sequence of finite length. There is no updating of the database because the database is used just once to encode the entire source sequence. Let l be a fixed positive integer. The source sequence is $x = (x_0, \ldots, x_{l-1})$. There is just one source segment, namely, $x^{(1)} = x$. The infinite database $y^{(1)}$ is given as some sequence $(\ldots, x_{-3}, x_{-2}, x_{-1})$. The codeword $b^{(1)}$ is taken to be the binary expansion of smallest positive integer $i \geq l$ such that $(x_{-i}, \ldots, x_{-i+l-1}) = x$. (We suppose that there is a substring of $y^{(1)}$ equal to x.) Assuming that the source samples and the database samples are drawn from a common memoryless model, the compression rate in Example 3 is a random variable R_l which is known to converge almost surely as $l \to \infty$ to the entropy rate of the model.

Although the string-matching algorithms of Examples 1-3 are different, we have seen that their performances are related to one another (at least for memoryless models). This is no coincidence. It is reasonable that as the fixed database in Example 1 becomes large, the compression performance in Example 1 can be related to the compression performance of the infinite database algorithm in Example 3. Similarly, since in Example 2 we have a database growing without bound, it is reasonable that the compression performance in Example 2 can be related to the compression performance in Example 3.

We are now ready to describe the approach to the analysis of string-matching data compression algorithms that shall be followed in this paper. In our approach, we shall determine random source and database models for which one can determine the limiting behavior as $l \to \infty$ of the compression rate R_l of the Example 3 string-matching algorithm for lossless coding, as well as the behavior of the extension of the Example 3 algorithm to lossy coding. This approach makes sense for two reasons. The first reason is that algorithms of the Example 3 type are the simplest of all string-matching algorithms to analyze. The second reason is that for many source/database models, the compression performance of string-matching algorithms more complicated than that of Example 3 can be related to the performance of the Example 3 algorithm, as indicated in the preceding paragraph.

In the next section, we give a precise statement of the results concerning string matching data compression that shall be established in this paper, followed by a discussion relating our results to the results of previous workers in this area.

II. STATEMENT AND DISCUSSION OF MAIN RESULTS

II-A. Concept of Source and Database

In describing databases and sources, we shall make use of deterministic or random sequences of the form $x = (x_i : i \in D)$, where the domain D is either the set of all integers, the set of negative integers, or the set of nonnegative integers. For such a sequence x, if $m, n \in D$ satisfy $m < n$, we write x_m^n to designate the vector $(x_m, x_{m+1}, \ldots, x_n)$. We fix throughout the paper a finite nonempty set A which shall serve as our source alphabet and our database alphabet. Also, unless otherwise specified, the functions "log" and "exp" are understood throughout to be to the base 2.

In this paper, the terminology "source" refers to a forward unilateral process $X = (X_i : i \geq 0)$ in which each source sample X_i is a random variable taking its values in the set A. The terminology "database" refers to a backward unilateral process $Y = (Y_i : i < 0)$ in which each database sample Y_i is a random variable taking its values in the set A. If X is a source and Y is a database, our string-matching data compression methodology involves the encoding of a block of samples in X by finding a match for this block in Y. For this methodology to make sense, we shall implicitly always assume that the source X and its accompanying database Y are defined on the same probability space. Given a source X and a database Y, the pair (Y, X) is intended to denote the bilateral process $(\ldots, Y_{-3}, Y_{-2}, Y_{-1}, X_0, X_1, X_2, \ldots)$ consisting of the database samples together with the source samples. A source X or a database Y or the pair (Y, X) is said to be *memoryless* if it consists of independent and identically distributed samples.

In this paper, a database Y shall always be a stationary process which will satisfy a certain mixing condition that insures that blocks that are windowed far apart from one another in Y are approximately statistically independent. (This approximate independence is needed in proofs that shall appear subsequently.) In this paragraph, we explain the type of mixing condition that shall be imposed. Suppose Y is a stationary database. Let $(\Omega, \mathcal{F}, \mu)$ be the probability space on which Y is defined. For every pair of negative integers m, n for which $m < n$, let $\mathcal{F}_m^n$ denote the subsigmafield of $\mathcal{F}$ generated by the random vector Y_m^n. For each positive integer k, let $\phi(k)$ be the number defined by

$$\phi(k) = \sup\{|\mu(G|F) - \mu(G)| : (G, F) \in \Phi_k\}, \tag{2.1}$$

where Φ_k is the set of all pairs (G, F) in which, for some triple (r, s, t) of negative integers satisfying $r \leq s \leq t - k$, G is an event of positive probability in the sigmafield $\mathcal{F}_r^s$ and F is an event of positive probability in the sigmafield $\mathcal{F}_t^{-1}$. The numbers $\phi(k), k \in Z$, are called the *mixing coefficients* of the database Y. The mixing condition needed for our proofs is the condition that the mixing coefficients $\{\phi(k)\}$ of the database Y be summable (i.e., satisfy $\sum_k \phi(k) < \infty$). This condition implies ergodicity of the database, but it is a stronger condition than ergodicity. As discussed in [14], here are some conditions under which a database Y has summable mixing coefficients:

(b.1) The database Y is a memoryless process.

(b.2) The database Y is a stationary, irreducible, aperiodic Markov chain.

(b.3) There is a finite-alphabet process $(U_i : i < 0)$ which forms a stationary, irreducible, aperiodic Markov chain, and a function f such that the process $(f(U_i) : i < 0)$ has the same distribution as Y.

It should be remarked that the assumptions (b.1) and (b.2) are special cases of assumption (b.3); we stated assumptions (b.1) and (b.2) separately for emphasis.

In lossless or lossy string-matching data compression, given a source X and a database Y, one assumes that the database Y is known to both encoder and decoder; one then encodes the source vector X_0^{l-1} for large l by encoding a positive integer i for which Y_{-i}^{-i+l-1} is a substring of Y providing a suitably good match to the source vector. In lossless compression, the matching substring Y_{-i}^{-i+l-1} must be equal to the source vector X_0^{l-1}, whereas in lossy compression, the matching substring Y_{-i}^{-i+l-1} must satisfy $\rho_l(Y_{-i}^{-i+l-1}, X_0^{l-1}) \leq D$, where ρ_l is a given distortion measure and D is a given distortion level.

In Section II-B which follows, we state our main result concerning lossless string-matching data compression (Theorem 1). Subsequently, in Section II-C, we state our main result concerning lossy string-matching data compression (Theorem 2).

II-B. Lossless String-Matching Data Compression

In a string-matching data compression algorithm based on a source X and a database Y, segments of the source X are encoded by matching them to substrings of the database Y. In lossless compression, the match must be exact. In order to insure that a matching substring can be found, we will need to impose a condition on source and database. Accordingly, we say that the source X and the database Y are *compatible* if for every $l > 0$ and every string $x \in A^l$ for which $\mathrm{Prob}[X_0^{l-1} = x] > 0$, it is almost surely true that $Y_{-i}^{-i+l-1} = x$ for some $i \geq l$.

Suppose the source X and the database Y are compatible. For each positive integer l, we define the random variable $N_l(Y, X)$ to be equal to the smallest $i \geq l$ such that $Y_{-i}^{-i+l-1} = X_0^{l-1}$. The functions $N_l(Y, X), l > 0$, shall be called the lossless string-matching functions induced by X, Y. The lossless string-matching function $N_l(Y, X)$ is important because anyone knowing the database Y can recover the source vector X_0^{l-1} upon learning the value of $N_l(Y, X)$. This insight allows us to construct a lossless string-matching data compression algorithm (Algorithm 1 below) of the type given in Example 3 of the introduction. Our implementation of Algorithm 1 employs the Elias integer code [2], which is a well known lossless coding technique via which each positive integer is converted into a binary codeword from a prefix set of codewords.

Algorithm 1. Fix a compatible source X and database Y. Fix a positive integer l. It is desired to encode the source vector X_0^{l-1} given that encoder and decoder have knowledge of the database Y. The Algorithm 1 encoder encodes the vector X_0^{l-1} into the binary codeword which is the Elias codeword for the random integer $N_l(Y, X)$. From this binary codeword, the Algorithm 1 decoder determines the value of the function $N_l(Y, X)$; this value enables the decoder to locate the window in the database Y where the source vector X_0^{l-1} may be found.

We let $R_l(Y, X)$ denote the random variable which is the compression rate of Algorithm 1, defined as the length of the codeword for X_0^{l-1} divided by l. It can be easily shown that the sequence $\{l^{-1} \log N_l(Y, X) : l > 0\}$ converges almost surely to a finite limit C if and only if the sequence $\{R_l(Y, X) : l > 0\}$ does also. (To demonstrate this, one uses the fact that the Elias codeword for the integer i is of length $\log i + O(\log \log i)$.) Thus, the problem of determining the asymptotic behavior of the sequence of compression rates $\{R_l(Y, X) : l > 0\}$ in Algorithm 1 is equivalent to the problem of determining the asymptotic behavior of the sequence of lossless string-matching functions $\{l^{-1} \log N_l(Y, X) : l > 0\}$.

To conclude this subsection, we state our main result on lossless string-matching data compression. For the proof of this result, please see [13]. In the following (and elsewhere), if X is a stationary source, we let $H(X)$ denote the entropy rate of the source X.

Theorem 1 *Let X be a source and let Y be a database. We suppose that Y is stationary with summable mixing coefficients, that the source X has the same finite-dimensional distributions as Y, and that X and Y are statistically independent. Then, Y and X are compatible and as $l \to \infty$,*

$$\frac{\log N_l(Y,X)}{l} \to H(X) \text{ almost surely.}$$

As a consequence, the compression rate $R_l(Y,X)$ in Algorithm 1 converges almost surely to $H(X)$.

II-C. Lossy String-Matching Data Compression

In this subsection, we are concerned with lossy data compression relative to a single-letter fidelity criterion $\rho = \{\rho_n : n = 1, 2, \ldots\}$. The distortion measures $\{\rho_n\}$ comprising the fidelity criterion ρ are assumed to satisfy the following requirements: (c.1) If $x, y \in A$, $\rho_1(x,y) \geq 0$; (c.2) for each x in A, there exists y in A such that $\rho_1(x,y) = 0$; and (c.3) if $n \geq 2$, $x = (x_1, x_2, \ldots, x_n) \in A^n$, and $y = (y_1, y_2, \ldots, y_n) \in A^n$, then $\rho_n(x,y) = n^{-1}\sum_i^n \rho_1(x_i, y_i)$. Let X be a source and let Y be a database. We say that the source X and the database Y are ρ-compatible if for every $l > 0$, every $D > 0$, and every string $x \in A^l$ for which $\text{Prob}[X_0^{l-1} = x] > 0$, it is almost surely true that $\rho_l(Y_{-i}^{-i+l-1}, x) \leq D$ for some $i \geq l$.

Suppose a source X and a database Y are ρ-compatible. Let l be a positive integer, and let $D > 0$. Then, almost surely, in one of the time windows $[-i, -i+l-1], i = l, l+1, \ldots$ there appears a substring Y_{-i}^{i+l-1} of the database Y that provides a good match for the source string X_0^{l-1} in the sense that

$$\rho_l(Y_{-i}^{-i+l-1}, X_0^{l-1}) \leq D$$

holds. We define a random variable $N_l(Y,X,D)$ giving us the position of the first such window in which a good match is found. Specifically, $N_l(Y,X,D)$ is defined to be equal to the smallest $i \geq l$ such that the above inequality holds. The functions $\{N_l(Y,X,D) : l > 0, D > 0\}$ shall be called the lossy string-matching functions induced by X, Y. Using the lossy string-matching function $N_l(Y,X,D)$, we can define the following lossy string-matching data compression algorithm.

Algorithm 2. We are given a source X and a database Y which are ρ-compatible. Fix a positive integer l, and fix the desired distortion level $D > 0$. The Algorithm 2 encoder and decoder employ the database Y and operate as follows. The encoder encodes the vector X_0^{l-1} into the binary codeword which is the Elias codeword for the integer $N_l(Y,X,D)$. From this binary codeword, the decoder determines the value i of the function $N_l(Y,X,D)$; the decoder output is then Y_{-i}^{-i+l-1}.

We let $R_l(Y,X,D)$ denote the random variable which is the compression rate of Algorithm 2, defined as the length of the codeword for X_0^{l-1} divided by l. As discussed in our earlier subsection on lossless compression, the asymptotic behavior of the sequence of compression rates $\{R_l(Y,X,D) : l > 0\}$ is determined by the asymptotic behavior of the sequence of string-matching functions $\{l^{-1}\log N_l(Y,X,D) : l > 0\}$.

Before stating our main result on lossy string-matching data compression, Theorem 2(its proof can be found in [13]), we present some more definitions and notation. The relative entropy distance $D(p||q)$ from a probability distribution $p = (p_i : i \in B)$ on a finite set B to a probability distribution $q = (q_i : i \in B)$ on that same set, is defined by

$$D(p||q) = \sum_{i \in B} p_i \log(\frac{p_i}{q_i}),$$

where the term $p_i \log(\frac{p_i}{q_i})$ is taken to be 0 if $p_i = 0$, and is taken to be ∞ if $p_i > 0, q_i = 0$.

Let X be a source and let Y be a database. The notation P shall denote the distribution of X and Q shall denote the distribution of Y. For $n \geq 1$, we let P_n and Q_n denote the distributions of X_0^{n-1} and Y_{-n}^{-1}, respectively.

Again, let X be a source and let Y be a database. Let $D > 0$. Define

$$R_0(P_1, Q_1, D) \triangleq \inf_U [I(X_0; U) + D(P_U||Q_1)],$$

where the infimum is taken over all A-valued random variables U that are jointly distributed with X_0 and satisfy $E\rho_1(X_0, U) \leq D$, and where P_U denotes the distribution of U.

Let $k \geq 1$ and let $D > 0$. For an arbitrary source X and database Y, define

$$R_k(P_{k+1}, Q_{k+1}, D) \triangleq$$
$$\inf_{U_1,\cdots,U_{k+1}} \{I(X_0^k; U_{k+1}|U_1^k) + \sum_{u \in A^k} \Pr\{U_1^k = u\} \times$$
$$D(P_{U_{k+1}}(\cdot|U_1^k = u)||P_{Y_{-1}}(\cdot|Y_{-k-1}^{-2} = u))\}.$$

The above infimum is taken over all A^{k+1}-valued random vectors $(U_1, \cdots, U_{k+1})$ that are jointly distributed with X_0^k and satisfy the properties:

(d.1) $\{(X_i, U_{i+1})\}_{i=0}^k$ is stationary in the sense that $\{(X_i, U_{i+1})\}_{i=0}^{k-1}$ and $\{(X_i, U_{i+1})\}_{i=1}^k$ have the same distribution.

(d.2) $E\rho_1(X_k, U_{k+1}) \leq D$.

Furthermore, $P_{U_{k+1}}(\cdot|U_1^k = u)$ denotes the conditional probability distribution of U_{k+1} given $U_1^k = u$, and $P_{Y_{-1}}(\cdot|Y_{-k-1}^{-2} = u)$ denotes the conditional probability distribution of Y_{-1} given $Y_{-k-1}^{-2} = u$.

Finally, if a source X is stationary, let $R_X(D)$ denote the rate distortion function of the source X relative to the fidelity criterion ρ, evaluated at D.

Theorem 2 *Let X be a source and let Y be a database. Let $D > 0$. Then each of the following hold.*

(i) *Let (Y, X) be a memoryless process. Then X and Y are ρ-compatible and as $l \to \infty$,*

$$\frac{1}{l} \log N_l(Y, X, D) \overset{a.s.}{\to} R_0(P_1, P_1, D).$$

As a consequence, the compression rate $R_l(Y, X, D)$ in Algorithm 2 converges almost surely to $R_0(P_1, P_1, D)$.

(ii) *Let $k \geq 1$, let (Y, X) be a stationary k-th order Markov process with positive transition probabilities and let $R_X(D) > 0$. Then X and Y are ρ-compatible and as $l \to \infty$,*

$$\frac{1}{l} \log N_l(Y, X, D) \overset{a.s.}{\to} R_k(P_{k+1}, P_{k+1}, D).$$

As a consequence, the compression rate $R_l(Y, X, D)$ converges almost surely to $R_k(P_{k+1}, P_{k+1}, D)$.

(iii) *Let Y be a memoryless process with Q_1 strictly positive on A, let X be a stationary ergodic process, and let Y and X be statistically independent. Then X and Y are ρ-compatible and as $l \to \infty$,*

$$\frac{1}{l} \log N_l(Y, X, D) \overset{a.s.}{\to} R_0(P_1, Q_1, D).$$

As a consequence, the compression rate $R_l(Y, X, D)$ converges almost surely to $R_0(P_1, Q_1, D)$.

(iv) *Let Y be a stationary finite-order Markov process with positive transition probabilities, let X be a stationary ergodic finite-order Markov process, and let Y and X be statistically independent. Let $k =$ the maximum of the orders of Y and X. Then X and Y are ρ-compatible and as $l \to \infty$,*

$$\frac{1}{l} \log N_l(Y, X, D) \overset{a.s.}{\to} R_k(P_{k+1}, Q_{k+1}, D).$$

As a consequence, the compression rate $R_l(Y, X, D)$ converges almost surely to $R_k(P_{k+1}, Q_{k+1}, D)$.

Remark 1 Theorem 2 applies to the lossy string matching functions $\{N_l(Y, X, D) : l > 0, D > 0\}$. Some other workers in lossy string-matching data compression have employed a slightly different notion of string matching function which we now discuss. Given a source X and a database Y, define $\tilde{N}_l(Y, X, D)$ to be the smallest positive integer i such that $\rho_l(Y_{-il}^{-(i-1)l-1}, X_0^{l-1}) \leq D$. The difference between the string-matching functions $N_l(Y, X, D)$ and $\tilde{N}_l(Y, X, D)$ is as follows. In defining $N_l(Y, X, D)$, we examine the "sliding" time windows $[-i, -i+l-1], i = l, l+1, l+2, \cdots$ in order to find the first such window where a substring of the database Y appears that provides the desired match to the source string X_0^{l-1}. In defining $\tilde{N}_l(Y, X, D)$, one examines instead the smaller set of "nonoverlapping" windows $[-i, -i+l-1], i = l, 2l, 3l, \cdots$. One can modify Algorithm 2 to obtain an Algorithm $\tilde{2}$; simply use the string-matching function $\tilde{N}_l(Y,X,D)$ for encoding instead of the string-matching function $N_l(Y,X,D)$. Denote the compression rate for Algorithm $\tilde{2}$ as $\tilde{R}_l(Y,X,D)$. We remark that Theorem 2 remains valid if in all convergence statements we replace $N_l(Y,X,D)$ and $R_l(Y,X,D)$ with $\tilde{N}_l(Y,X,D)$ and $\tilde{R}_l(Y,X,D)$, respectively. One can easily modify our proof of Theorem 2 to show this, but we shall not carry out the details of this task in this paper.

II-D. Implications for Source Coding

We discuss the implications of Theorem 2 for lossy source code design. We first discuss the case of memoryless databases and then the case of finite-order Markov databases.

Memoryless databases. Consider a source X and database Y in which the source X is stationary ergodic, the database Y is memoryless, and X,Y are statistically independent. Fix the desired distortion level $D > 0$. Theorem 2 tells us that if we encode the source vector X_0^{l-1} using Algorithm 2, the asymptotic compression rate as the number of source samples l tends to infinity is almost surely equal to the number $R_0(P_1,Q_1,D)$. Algorithm 2 is therefore optimal if and only if $R_0(P_1,Q_1,D) = R_X(D)$. We consider the following question. Suppose we fix the source X (which implies that the distribution P_1 of the source samples is fixed). Can we choose the distribution Q_1 of the database samples so that Algorithm 2 is asymptotically optimal for the source X, and, if so, how is Q_1 selected? To answer this question, we employ the first order approximation $R_0(P_1,D)$ to the rate distortion function $R_X(D)$ given by

$$R_0(P_1,D) = \inf_U I(X_0;U) \,,$$

where the infimum is taken over the same range as used in defining $R_0(P_1,Q_1,D)$. We show that Q_1 can be chosen to make Algorithm 2 optimal if and only if the source X is such that $R_0(P_1,D) = R_X(D)$. (This equality holds for every memoryless source, whereas the equality will fail for a wide class of sources with memory.) Comparing the definitions of $R_0(P_1,D)$ and $R_0(P_1,Q_1,D)$, it is clear that

$$R_0(P_1,Q_1,D) \geq R_0(P_1,D) \geq R_X(D). \quad (2.2)$$

no matter how Q_1 is chosen. Suppose the source is such that $R_0(P_1,D) > R_X(D)$. Then, from (2.2), we conclude that $R_0(P_1,Q_1,D) > R_X(D)$ for all Q_1 and therefore Q_1 can never be chosen to make Algorithm 2 optimal for the given source X. Suppose now that the source X is such that $R_0(P_1,D) = R_X(D)$. Then, Algorithm 2 is optimal provided one chooses Q_1 to be the solution of the equation $R_0(P_1,Q_1,D) = R_0(P_1,D)$. Solving this equation, one sees that Q_1 should be chosen so that $Q_1 = P_{U^*}$, where P_{U^*} is the distribution of the random variable U^* which achieves the infimum in the definition of $R_0(P_1,D)$.

Example 1. We suppose that the source X is memoryless, that $A = \{0,1\}$, and that the distortion measure ρ_1 is Hamming distortion. This is the binary memoryless source (BMS) model. We require that the database Y must also be memoryless, and that X,Y are statistically independent. The distribution of each source sample is $P_1 = (P_1(0),P_1(1))$, and the distribution of each database sample is $Q_1 = (Q_1(0),Q_1(1))$. The parameter $P_1(1)$ fixes the source, and the parameter $Q_1(1)$ fixes Algorithm 2 via the database. We address the following two questions. Given $P_1(1)$, how should $Q_1(1)$ be chosen so that the resulting Algorithm 2 is optimal for the source governed by $P_1(1)$? Can $Q_1(1)$ be chosen so that Algorithm 2 determined by $Q_1(1)$ is a universal code for the class of binary memoryless sources in the sense of being optimal for every such source? The answer to the first question is to choose $Q_1(1)$ so that Q_1 is the output distribution of the test channel achieving the infimum in the definition of $R_0(P_1,D)$. Assuming $0 < D < P_1(1) \leq 1/2$, this means we select $Q_1(1)$ as follows:

$$Q_1(1) = \frac{P_1(1) - D}{1 - 2D} \quad (2.3)$$

(See [1], page 343.) To answer the second question, note that each value of $Q_1(1)$ determines a unique value of $P_1(1)$ in (2.3). Hence, no matter how we choose $Q_1(1)$, the resulting Algorithm 2 will be optimal for at most one binary memoryless source, and hence Algorithm 2 fails to be universal in a rather strong way.

Example 2. Let the source X and the database Y be such that the process (Y,X) is memoryless, and let the alphabet A be binary. Theorem 2 tells us that the asymptotic compression rate for Algorithm 2 is almost surely equal to the number $R_0(P_1, P_1, D)$, where $P_1 = (P_1(0), P_1(1)) = (p, 1-p)$ is the common distribution for source samples and database samples. We assume that $0 < p < 1/2$. The following equations give a closed form expression for $R_0(P_1, P_1, D)$ that is obtainable using the Lagrange multiplier method:

$R_0(P_1, P_1, D) = 0$ if $D \geq 2p(1-p)$; otherwise

$$R_0(P_1, P_1, D) = pD((x, 1-x)||P_1) + (1-p)D((y, 1-y)||P_1)\ ,$$

where

$$x \triangleq \frac{-(D - p - 2pD + 3p^2)}{2p(1-2p)} + \frac{\sqrt{(D - p - 2pD + 3p^2)^2 + 4p^3(1-D)(1-2p)}}{2p(1-2p)}$$

and

$$y \triangleq (D - p + px)/(1-p)\ .$$

On the other hand, the rate distortion function $R_X(D)$ is given by

$$R_X(D) = \begin{cases} 0 & \text{if } D \geq p \\ h(p) - h(D) & \text{otherwise,} \end{cases}$$

where $h(p) = -p \log p - (1-p)\log(1-p)$. Since $2p(1-p) > p$, the functions $R_0(P_1, P_1, D)$ and $R_X(D)$ do not coincide.

Databases with Markov memory. Let k be a positive integer. Consider source X and database Y in which the source X is fixed to be a stationary ergodic Markov chain of order $\leq k$, and X, Y are statistically independent. Suppose the database Y is to be chosen, if possible, so that the resulting Algorithm 2 will be asymptotically optimal for the given source X. We require that the database Y is to be chosen from the class of Markov chains of order $\leq k$ having positive transition probabilities. For each such database, the resulting Algorithm 2 will be asymptotically optimal for the source X if and only if $R_k(P_{k+1}, Q_{k+1}, D) = R_X(D)$. One can define an upper bound $R_k(D)$ to $R_X(D)$, dependent on the source but not the database, such that there will exist a database for which Algorithm 2 is optimal for the source if and only if $R_k(D) = R_X(D)$, and if this condition is satisfied, one can characterize the database Y that makes Algorithm 2 optimal. This result is not of much practical significance at the present time due to the paucity of results concerning the form of the rate-distortion function for sources with Markov memory; also, the result is messy to derive. Consequently, we shall not derive this result here.

II-E. Previous Results

We discuss the relation of our work to the works of previous authors on string-matching data compression. First, we consider previous results on lossless string-matching data compression. Wyner and Ziv [10] proved the following result.

Result 1 *Let X be a source and let Y be a database.*

(i) *Suppose the bilateral process (Y, X) is stationary and ergodic. Then as $l \to \infty$*

$$\frac{1}{l} \log N_l(Y, X) \to H(X) \text{ in probability.} \tag{2.4}$$

(ii) *Suppose that X is a stationary irreducible Markov chain, that Y has the same finite-dimensional distributions as X, and that X, Y are statistically independent. Then (2.4) holds as $l \to \infty$.*

Part (i) of Result 1 was extended to the case of almost sure convergence by Ornstein and Weiss [6]; Szpankowski [9] obtained this same result by a different technique. Nobel and Wyner [5] extended part (ii) of Result 1 to the case when Y (and therefore X) is a stationary process whose mixing coefficients $\{\phi(k)\}$ are $O(1/k^{1+\delta})$ for some $\delta > 0$, and Shields [7] extended this same result, obtaining almost sure convergence for finite-state source models. Our Theorem 1 extends the just-quoted results of Nobel-Wyner and Shields.

We now discuss previous results on lossy string-matching data compression. Steinberg and Gutman [8] established the following lossy string-matching data compression result.

Result 2 *Let X be a source and let Y be a database. Let $D > 0$.*

(i) *Suppose that the bilateral process (Y, X) is stationary and totally ergodic, that ρ_1 is a metric on A, and that $R_X(D/2) > 0$. Let $\epsilon > 0$. Then as $l \to \infty$*

$$\Pr\{\frac{1}{l}\log \tilde{N}_l(Y, X, D) \geq R_X(D/2) + \epsilon\} \to 0. \tag{2.5}$$

(ii) *Suppose X is a stationary irreducible aperiodic Markov chain, that Y has the same finite-dimensional distributions as X, and that X, Y are statistically independent. Suppose $R_X(D/2) > 0$, and let $\epsilon > 0$. Then (2.5) holds as $l \to \infty$.*

Result 2 states that $R_X(D/2)$ is an upper bound on the asymptotic performance of the sequence of string-matching functions $\{l^{-1}\log \tilde{N}_l(Y, X, D) : l > 0\}$. Steinberg and Gutman [8] also obtained a tighter upper bound than $R_X(D/2)$ in the case of a memoryless source X with Hamming distortion measure ρ_1. (The exact formula for this upper bound is not important to us here.) Koga and Arimoto [3] also obtained an upper bound in the memoryless source case. Their upper bound takes the form $R_X(D) + D(P_1||P^*)$, where P^* is the output distribution on A of the optimal test channel that yields $R_X(D)$. Luczak and Szpankowski [4] proved the convergence in probability of the sequence $\{l^{-1}\log N_l(Y, X, D) : l > 0\}$ for (Y, X) stationary subject to a mixing condition on Y; we obtain the almost sure convergence result under less restrictive assumptions in part (i) of Lemma 1 of the present paper. Furthermore, the papers [8] [3] [4] did not report much success in characterizing the precise asymptotic performance of sequences of form $\{l^{-1}\log \tilde{N}_l(Y, X, D) : l > 0\}$ or $\{l^{-1}\log N_l(Y, X, D) : l > 0\}$, although Luczak and Szpankowski [4] did show how to compute the limit of the latter sequence in the case of binary memoryless (Y, X). In view of the lossy string-matching results just quoted, we were motivated in our paper to try to obtain a formula for the almost sure limit of the sequence $\{l^{-1}\log N_l(Y, X, D) : l > 0\}$ for as wide a class of databases and sources as possible. The Koga-Arimoto upper bound expression was an inspiration to us in this effort; it suggested to us the quantities $R_0(P_1, Q_1, D)$ and $R_k(P_{k+1}, Q_{k+1}, D)$ which give the precise asymptotic performance of $\{l^{-1}\log N_l(Y, X, D) : l > 0\}$ in the memoryless and finite-order Markov source cases, respectively.

II-F. Methodology of Proof

In the subsequent sections, we shall prove our main results Theorems 1 and Theorem 2. The proof of Theorem 1 is straightforward in that it involves a sharpening of the proof techniques employed by Wyner and Ziv [10] and by those who extended the Wyner-Ziv lossless coding results. The proof of Theorem 2 is much more interesting in that it involves a methodology that can be used as a general tool for the analysis of lossy string-matching data compression algorithms. We devote this subsection to an explanation of this methodology.

For each $x \in A^n$, let $B(x, D)$ denote the set of all sequences $y \in A^n$ such that $\rho_n(x, y) \leq D$. The set $B(x, D)$ can be thought of as a ball centered at x with radius D.

Let X be a source and let Y be a database. Then $\{Q_n(B(X_0^{n-1}, D)) : n > 0\}$ is a well-defined sequence of random variables. Our methodology for showing the convergence of the sequence of lossy string-matching functions $\{l^{-1}\log N_l(Y, X, D) : l > 0\}$ is to show instead the convergence of the sequence $\{-l^{-1}\log Q_l(B(X_0^{l-1}, D)) : l > 0\}$. By means of a useful lemma stated at the end of this section, convergence of the latter sequence will imply convergence of the former sequence to the same limit, under certain restrictions on X and Y.

To obtain our lemma, we shall require a stronger mixing condition for databases than used previously. We define this condition here. Let Y be a stationary database. We define the *symmetric mixing coefficients* of Y to be the numbers $\{\bar{\phi}(k) : k \geq 1\}$ defined by

$$\bar{\phi}(k) \triangleq \sup\{\max(|\mu(F|G) - \mu(F)|, |\mu(G|F) - \mu(G)|)\},$$

where the supremum is taken over all pairs $(G, F) \in \Phi_k$, and Φ_k is the same set employed in (2.1). We shall say that Y has *summable symmetric mixing coefficients* if $\sum_k \bar{\phi}(k) < \infty$. We point out that the condition that Y have summable symmetric mixing coefficients is a stronger condition than the condition that Y have summable mixing coefficients. (This is because $\bar{\phi}(k) \geq \phi(k)$ for every k.) However, a database Y will have summable symmetric mixing coefficients under any of the conditions (b.1)-(b.3).

Lemma 1 *Let X be a source and let Y be a database. Let $D > 0$, and assume that the sequence $\{-l^{-1} \log Q_l(B(X_0^{l-1}, D)) : l > 0\}$ converges almost surely as $l \to \infty$ to a real constant R^*.*

(i) *Suppose that the bilateral process (Y, X) is stationary, that Y has summable symmetric mixing coefficients $\{\bar{\phi}(k)\}$, and that $R_X(D) > 0$. Then X and Y are ρ-compatible and as $l \to \infty$,*

$$\frac{1}{l} \log N_l(Y, X, D) \overset{a.s.}{\to} R^*. \qquad (2.6)$$

(ii) *Suppose that X is stationary ergodic, that Y is stationary with summable mixing coefficients $\{\phi(k)\}$, and that X, Y are statistically independent. Then X and Y are ρ-compatible and (2.6) holds as $l \to \infty$.*

In all instances in this paper in which we apply Lemma 1, the source X shall be stationary ergodic and the database Y shall be a stationary finite-order Markov chain with positive transition probabilities. Under these restrictions, the sequence $\{-l^{-1} \log Q_l(B(X_0^{l-1}, D)) : l > 0\}$ automatically converges almost surely to a constant, by the subadditive ergodic theorem. To deduce Theorem 2 from this convergence is then purely a matter of identifying this constant by means of an appropriate formula. However, in the future, databases Y more general than finite-order Markov might be found for which the limit of $\{-l^{-1} \log Q_l(B(X_0^{l-1}, D)) : l > 0\}$ exists. In such a case, one can at once deduce convergence of the lossy string-matching functions induced by X, Y, provided Y satisfies a mixing condition that will make Lemma 1 hold.

REFERENCES

[1] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* New York: Wiley, 1991.

[2] P. Elias, "Universal codeword sets and representations of the integers," *IEEE Trans. Inform. Theory,* vol. IT-21, pp. 194-203, 1975.

[3] H. Koga and S. Arimoto, "Asymptotic properties of algorithms of data compression with fidelity criterion based on string matching," *Proc. of IEEE ISIT'94*, Norway, Trondheim, p. 264.

[4] T. Luczak and W. Szpankowski, "A suboptimal lossy data compression based on approximate pattern matching," manuscript submitted for publication.

[5] A. Nobel and A. D. Wyner, "A recurrence theorem for dependent processes with applications to data compression," *IEEE Trans. Inform. Theory,* Vol. 38, pp. 1561–1563, 1992.

[6] D. S. Ornstein and B. Weiss, "Entropy and data compression schemes," *IEEE Trans. Inform. Theory,* Vol. 39, pp. 78–83, 1993.

[7] P. C. Shields, "Waiting times: positive and negative results on the Wyner-Ziv problem," *J. Theoretical Proba.,* Vol. 6, No. 3, pp. 499–519, 1993.

[8] Y. Steinberg and M. Gutman, "An algorithm for source coding subject to a fidelity criterion based on string matching," *IEEE Trans. Inform. Theory,* vol. IT-39, pp. 877–886, 1993.

[9] W. Szpankowski, "Asymptotic properties of data compression and suffix trees," *IEEE Trans. Inform. Theory,* vol. IT-39, pp. 1647-1659, 1993.

[10] A. D. Wyner and J. Ziv, "Some asymptotic properties of the entropy of a stationary ergodic data source with applications to data compression," *IEEE Trans. Inform. Theory,* Vol. IT-35, pp. 1250–1258, 1989.

[11] A. D. Wyner, "The sliding window Lempel-Ziv algorithm is asymptotically optimal," Proc. IEEE, Vol. 82, pp. 872–877, 1994.

[12] A. J. Wyner and A. D. Wyner, "Improved redundancy of a version of the Lempel-Ziv algorithm," *IEEE Trans. Inform. Theory,* Vol. IT-41, pp. 723-731, 1995.

[13] E.-H. Yang and J. C. Kieffer, "On the Performance of Data Compression Algorithms Based upon String Matching," *Submitted IEEE Trans. Inform. Theory* for publication.

[14] Z. Zhang and E.-H. Yang, "An on-line universal lossy data compression algorithm by continuous codebook refinement–Part II: optimality for ϕ-mixing source models," to appear in *IEEE Trans. Inform. Theory.*

ON GENERALIZED DIGITAL SEARCH TREES WITH APPLICATIONS TO A GENERALIZED LEMPEL-ZIV ALGORITHM

Wojciech Szpankowski*
Department of Computer Science
Purdue University
W. Lafayette, IN 47907
U.S.A.
`spa@cs.purdue.edu`

Jing Tang
Department of Mathematics
Purdue University
W. Lafayette, IN 47907
U.S.A.
`tangj@math.purdue.edu`

Abstract

The goal of this research is twofold: (i) to analyze generalized digital search trees, and (ii) to derive the average profile (i.e., phrase length) of a generalization of the well known parsing algorithm due to Lempel and Ziv. In the generalized Lempel-Ziv parsing scheme, one partitions a sequence of symbols from a finite alphabet into phrases such that the new phrase is the longest substring seen in the past by at most $b-1$ phrases. The case $b=1$ corresponds to the original Lempel-Ziv scheme. As expected, such a scheme can be analyzed by considering a generalized digital search tree in which every node is capable of storing up to b strings. Such a digital tree is known as b-digital search tree. In this paper, we concentrate on investigating the depth of a randomly selected node in such a tree. We apply analytical techniques of analysis of algorithms to establish the mean, the variance and the limiting distribution of the depth. Based on these results, we infer the probabilistic behavior of a randomly selected phrase in the generalized Lempel-Ziv algorithm.

1. INTRODUCTION

The heart of several universal data compression schemes is the parsing algorithm due to Lempel and Ziv [16]. It partitions a sequence into phrases (blocks) of variable sizes such that a new block is the shortest substring not seen in the past as a phrase. For example, 11001010001000100011 is parsed into (1)(10)(0)(101)(00)(01)(000)(100)(11). This scheme is used in the UNIX `compress` command and in a CCITT standard for data compression for modems. But, the original Lempel-Ziv scheme does not work very well for sequences with long repeated symbols (i.e., in such a case a run-length code is more useful). For example, the sequence 0000000000 has the Lempel-Ziv code of length ten (and thus it requires the same number of bits as the original string!). To remedy this situation, Louchard and Szpankowski [10] introduced a generalization of the Lempel-Ziv parsing scheme. It parses a sequence into phrases such that the next phrase is the longest phrase seen in the past by *at most* $b-1$ phrases ($b=1$ corresponds to the original Lempel-Ziv algorithm). A data compression code for this new algorithm may consists of pairs of numbers: one being a pointer to the previous occurrence of the prefix of the phrase, and the second number is either empty or the last bit of the phrase in the case it is the bth phrase. For example, the previous string 0000000000 is parsed into (0)(0)(00)(00)(000)(0) for $b=2$ and requires only eight bits to code it.

Our goal is to investigate the probabilistic behavior of a typical phrase length. As already observed in Louchard and Szpankowski [9] (cf. [5]), one can analyze the Lempel-Ziv algorithm by exploring a digital search tree (cf. [1, 7, 11]). In the case of the

---

*This research was partially supported by NSF Grants NCR-9206315, NCR-9415491, and CCR-9201078, and NATO Collaborative Grant CRG.950060.

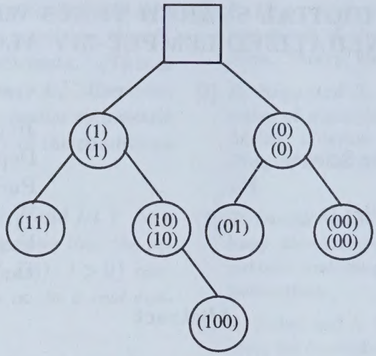


Figure 1: A 2-digital search tree representation of the generalized Lempel-Ziv parsing for the string 1100101000100010011

generalized Lempel-Ziv, we must consider a generalization of digital trees called *b-digital search tree*, or shortly b-DST (cf. [2, 11]) which is built from a fixed number, say m, of strings over a binary alphabet $\Sigma = \{0,1\}$ (extension to a finite alphabet is easy). This tree is constructed as follows: The first b strings are stored in the root. The remaining $m-b$ strings are stored in an available space in a node which is not full, i.e., containing less than b strings. The search for an available space follows the prefix structure of a string. The rule is simple: if the next symbol in a string is ”1“ we move to the right, otherwise move to the left until either we find a node with less than b strings or, if all nodes are full on this path, we create a new node. The details of such a construction can be found in [2, 7, 11]. To represent the generalized Lempel-Ziv scheme, we only must append the string to be parsed with b empty strings that are stored in the root. All other phrases are stored in internal nodes. When a new phrase is created, the search starts at the root and proceeds down the tree as directed by the input symbols of the string exactly in the same manner as in the digital tree construction untill either we find a node with less than b phrases or we need to create a new node. In Figure 1 we show a 2-DST constructed from the sequence 1100101000100010011. Observe that for a fixed length string, the number of nodes in the associated digital tree is random.

Digital trees appear in a variety of computer and communication applications including searching, sorting, dynamic hashing, codes stack communication protocols, and data compression as mentioned above (cf. [1, 2, 7, 8, 11, 9, 10, 5]). Although there is some unpleasant dependency between consecutive phrases in the Lempel-Ziv scheme, Louchard and Szpankowski [9] proved that this dependency is not too strong and one can infer the probabilistic behavior of the length of a randomly selected phrase from the depth of a randomly selected node in a b-DST built from *fixed* number of nodes. Therefore, in this conference version of the paper, we concentrate on studying b-digital search trees with fixed number, say m, strings.

We consider a digital tree in a probabilistic framework in which all m strings are statistically independent, and every string is generated according to the Bernoulli model. That is: *symbols are generated in an independent manner with ”0“ and ”1“ occurring respectively with probability p and $q = 1-p$.* If $p = q = 0.5$, the the Bernoulli model is called *symmetric*, otherwise it is *asymmetric.*

Our results can be summarized as follows: For the b-digital search tree we establish the mean, the variance and the limiting distribution of the depth in the asymmetric Bernoulli model. In particularly, we prove that the depth appropriately normalized is normally distributed. We also show that the same is true for a randomly selected phrase in the generalized Lempel-Ziv scheme. We establish these results in a uniform manner by a novel technique that belongs to the toolkit of the analytical analysis of algorithms. More precisely, it was already observed by Flajolet and Richmond [2] that b-digital trees are harder to analyze than the regular ($b = 1$) digital search tree. We further observe that the asymmetric model considered in this paper adds a new hardship compared to the symmetric model analyzed by Flajolet and Richmond [1]. In particular, in the asymmetric model one cannot solve exactly functional equations involved (since the solution is no any longer a harmonic sum), and one must restore to asymptotic solutions through poissonization, singularity analysis, Mellin transform, and depoissonization. In addition, we had to develop a technique to estimate some constants (of Mellin transform) occuring in the final solution. These constants are somewhat important since they carry the information about the dependence of b on the depth and phrase length.

Digital search trees for $b = 1$ have been analyzed in the past in the case of *fixed* number of independent strings (cf. [1, 7, 8, 11, 14]). The situation is quite different when general b-digital search trees are considered. The first non-trivial analysis of the size of such trees in the symmetric Bernoulli model was proposed by Flajolet and Richmond [2]. The variance of the size and internal path length – still for the symmetric model – was discussed by Hubalek [4]. To the best of our knowledge, b-DST have not yet been analyzed for the asymmetric Bernoulli model. On the other hand, for the original Lempel-Ziv parsing algorithm ($b = 1$) mostly first-order properties have been investigated, with an exception of some recent works of Louchard and Szpankowski [9], and Jacquet and Szpankowski [6], and there is no analysis of the generalized Lempel-Ziv scheme except preliminary discussion of [10].

2. MAIN RESULTS

We consider a b-digital tree built over m statistically independent words. Let D_m^i be the depth of the i-th string in a b-DST, that is, the length of a path from the root to a node containing the ith string. In a variety of applications, one is interested in the *typical* depth D_m that is defined as the depth of a randomly selected string, that is,

$$P\{D_m = k\} = \sum_{i=1}^{m} \frac{P\{D_m^i = k\}}{m} \tag{1}$$

As argued in Louchard and Szpankowski [9], the depth D_m is closely related to the length of a randomly selected phrase in the generalized Lempel-Ziv parsing scheme. We denote it as D_n^{LZ} where n is the length of the string to be parsed. Our goal is to study moments and distribution of D_m and D_n^{LZ}, and their dependence upon the parameter b.

As mentioned in the introduction, in this conference version of the paper we mostly concentrate on the depth D_m in a b-DST built over a fixed number of independent strings generated according to an asymmetric Bernoulli model (with "0" and "1" occurring respectively with probability p and $q = 1 - p$,). Let B_m^k be the average number of items (strings) on level k of a randomly built b-digital search tree. It is to observe that $\Pr\{D_m = k\} = B_m^k/m$ (cf. [7, 14]), thus one can alternatively study the average B_m^k which is further called *the average profile.* Let $B_m(u) = \sum_{k\geq 0} B_m^k u^k$ be its generating function. A simple analysis of the digital search tree reveals (cf. [9]) that $B_m(u)$ satisfies the following recurrence:

$$B_{m+b}(u) = b + u \sum_{i=0}^{m} \binom{m}{i} p^i q^{m-i} \left(B_i(u) + B_{m-i}(u)\right) \tag{2}$$

with initial conditions

$$B_i(u) = i \qquad \text{for} \qquad i = 0, 1, \ldots, b \tag{3}$$

For $b = 1$ the above recurrence can be solved as shown in [13] (cf. [9]). Unfortunately, for $b > 1$ the recurrence becomes too complicated to be solved exactly. This was already noted by Flajolet and Richmond [2] who developed a special technique to deal with such recurrences for $b > 1$. Unfortunately again, the technique of [2] was designed for the symmetric Bernoulli model, and becomes very intricate for the asymmetric Bernoulli models (due to the fact that some sums occurring in the solution of (2) cease to become harmonic sums in the asymmetric case).

In view of this, we approach the general recurrence (2) from a new "angle". First of all, we "poissonize" the model, that is, we introduce the Poisson transform (or Poisson generating function) as

$$\tilde{B}(u, z) = \sum_{i=0}^{\infty} B_i(u) \frac{z^i}{i!} e^{-z}.$$

Then, the recurrence becomes a slightly more manageable functional equation, namely:

$$(1 + \frac{\partial}{\partial z})^b \tilde{B}(u, z) = b + u \left(\tilde{B}(u, pz) + \tilde{B}(u, qz)\right) \tag{4}$$

where $(1 + \frac{\partial}{\partial z})^b f(z) =^{\text{def}} \sum_{i=0}^{b} \binom{b}{i} \frac{\partial^i f(z)}{\partial z^i}$.

In the next section, we use the Mellin transform [3] and the depoissonization lemma [12, 6] to solve the above equation. We prove the following main result.

Theorem 1 (i) *Let D_m be the typical depth in a b-digital tree built over m statistically independent strings under the* **asymmetric** *Bernoulli model. Then*

$$\begin{aligned} ED_m &= \frac{1}{h_1} \log m + \frac{1}{h_1} \left(\frac{h_2}{2h_1} + \gamma - 1 - H_{b-1} - \Delta(b, p) + \delta(\log_2 m) \right) + O\left(\frac{\log m}{\sqrt{m}}\right) \\ \text{Var } D_m &= \frac{h_2 - h_1^2}{h_1^3} \log m + O(1) \end{aligned}$$

where $h_1 = -p \log p - q \log q$ is the entropy, $h_2 = p \log^2 p + q \log^2 q$, and $\gamma = 0.577...$ is Euler constant, while $H_{b-1} = \sum_{i=1}^{b-1} \frac{1}{i}$, $H_0 = 0$ is the harmonic sum. The constant $\Delta(b, p)$ can be computed as follows

$$\Delta(b, p) = \sum_{n=2b+1}^{\infty} \bar{f}_n \sum_{i=1}^{b} \frac{(i+1) b!}{(b-i)! n (n-1) \ldots (n-i-1)} \tag{5}$$

where $\bar{f}_n$ is given recursively by

$$\begin{cases} f_{m+b} = m + \sum_{i=0}^{m} \binom{m}{i} p^i q^{m-i} (f_i + f_{m-i}), \ \text{for } m > 0, \\ f_0 = f_1 = \ldots = f_b = 0, \\ \bar{f}_{m+b} = f_{m+b} - m > 0, \ m \geq b + 1. \end{cases}$$

Finally, $\delta(\log_2 m)$ is a fluctuating function with a small amplitude when $\log p/\log q$ is rational, and $\delta(\log_2 m) \equiv 0$ otherwise.

(ii) *Let $G_m(u)$ be the ordinary generating function of D_m (i.e., $G_m(u) = Eu^{D_m}$), $\mu_m = ED_m$, and $\sigma_m = \sqrt{\mathrm{Var}\ D_m}$. Then, for a complex τ such that $\Re(\tau) > 0$*

$$e^{-\tau\mu_m/\sigma_m} G_m(e^{\tau/\sigma_m}) = e^{\frac{\tau^2}{2}}\left(1 + O\left(\frac{1}{\sqrt{\log m}}\right)\right) \tag{6}$$

Thus, the limiting distribution of $\frac{D_m - \mu_m}{\sigma_m}$ is normal, and it converges in moments to the appropriate moments of the standard normal distribution. Also, there exist positive constants A and $\alpha < 1$ such that

$$\Pr\left\{\left|\frac{D_m - c_1 \log m}{\sqrt{c_2 \log m}}\right| > k\right\} \leq A\alpha^k \tag{7}$$

uniformly in k for large m.

The situation is similar, but *not* the same in the **Lempel-Ziv model** in which a sequence of fixed length n is parsed into phrases. Let M_n denote the number of phrases produced by the algorithm. Let also $D_n^{LZ}(i)$ be the length of the ith phrase in the Lempel-Ziv model, where $1 \leq i \leq M_n$. By the *typical phrase length D_n^{LZ}* we denote the length of a randomly selected phrase.

The typical depth D_n^{LZ} in the Lempel-Ziv model can be estimated as follows

$$\Pr\{D_n^{LZ} = k\} = \sum_{m=m_L}^{m_U} \Pr\{D_n^{LZ} = k | M_n = m\}\Pr\{M_n = m\} \tag{8}$$

where m_L and m_U are the lower and the upper bounds for the number of phrases. It is easy to see that there exist constants α_1 and α_2 such that $m_L = \alpha_1\sqrt{n/b} \leq M_n \leq \alpha_2(n/b)/\log_2(n/b) = m_U$. Indeed, the minimum number of phrases occurs only for two strings: either all zeros or all ones, and then $n = \sum_{i=1}^{M_n} D_n(i) \leq b\sum_{i=1}^{M'n} i$, hence the lower bound $m_L = \Omega(\sqrt{n/b})$. For the upper bound, we consider a complete binary tree with the internal path length equal to n. Naturally, the number of nodes in such a tree is $O((n/b)/\log_2(n/b))$.

According to (8), one needs to estimate the conditional probability $\Pr\{D_n^{LZ} = k | M_n = m\}$ in order to assess the distribution of D_n^{LZ}. It is tempting to assume that $\Pr\{D_n^{LZ} = k | M_n = m\} = \Pr\{D_m = k\}$ where the right-hand side refers to the depth in the digital tree model. But, this is *untrue* due to the fact that in the Lempel-Ziv model we consider *only* those digital search trees whose internal path length is fixed and equal to n. Clearly, this restriction affects the depth of a randomly selected phrase (think of a digital tree built from the string $11111\ldots111$ which is very skewed). Fortunately, Louchard and Szpankowski [9] proved for $b = 1$ (and one can adopt without any significant changes the proof to the case $b > 1$) that $\Pr\{D_n^{LZ} = k | M_n = m\} = (1 + O(\sqrt{\log n/n}))\Pr\{D_m = k\}$.

Thus, the following second main result follows:

Theorem 2 *Let D_n^{LZ} be the length of a randomly selected phrase in the generalized Lempel-Ziv parsing of a string of length n under the* **asymmetric** *Bernoulli model. Then*

$$ED_n^{LZ} \sim \frac{1}{h_1}\log(nh_1/\log n) \tag{9}$$

$$\mathrm{Var}\ D_n^{LZ} \sim \frac{h_2 - h_1^2}{h_1^3}\log(nh_1/\log n)\ . \tag{10}$$

Furthermore, the limiting distribution of $\frac{D_n^{LZ}-ED_n^{LZ}}{\sqrt{\mathrm{Var}D_n^{LZ}}}$ *is normal.*

In passing, we should point out that the symmetric Bernoulli model differs significantly from the asymmetric model due to the fact (which we prove in a journal version of our paper) that $\mathrm{Var}D_m = O(1)$. In particular, the limiting distribution does not exist in this case, but it can be approximated by an extreme distribution.

3. ANALYSIS

In this section, we only prove Theorem 1 part (i) delaying the remaining proofs to the journal version.

As defined in the previous section, $B_m(u)$ is the generating function of the average profile B_m^k. Observe that $B_m(1) = m$, and $ED_m = B'_m(1)/m$, and finally $B''_m(1)/m = E\{D_m(D_m-1)\} = \mathrm{Var}D_m - ED_m + (ED_m)^2$. Thus, $\mathrm{Var}\ D_m = B''_m(1)/m + B'_m(1)/m - (B'_m(1)/m)^2$. We will use the above formula to derive asymptotics of ED_m and $\mathrm{Var}\ D_m$ as $m \to \infty$.

Our approach is analytical, and as mentioned in the previous section, we first derive the mean and the second factorial moment in the Poisson model. For this we need the first and the second derivative with respect to u at $u = 1$ of $B(u,z)$ derived in (4). One obtains:

$$(1+\frac{\partial}{\partial z})^b \tilde{B}_u(u,z) = \left(\tilde{B}(u,pz) + \tilde{B}(u,qz)\right) + u\left(\tilde{B}_u(u,pz) + \tilde{B}_u(u,qz)\right) ,$$

$$(1+\frac{\partial}{\partial z})^b \tilde{B}_{uu}(u,z) = 2\left(\tilde{B}_u(u,pz) + \tilde{B}_u(u,qz)\right) + u\left(\tilde{B}_{uu}(u,pz) + \tilde{B}_{uu}(u,qz)\right) .$$

Let $\tilde{B}_u(1,z) = \widetilde{X}(z)$, $\tilde{B}_{uu}(1,z) = \widetilde{W}(z)$ are the Poisson mean and the Poisson second factorial moment. Then, from the above we find

$$(1+\frac{\partial}{\partial z})^b \widetilde{X}(z) = z + \widetilde{X}(pz) + \widetilde{X}(qz) , \tag{11}$$

$$(1+\frac{\partial}{\partial z})^b \widetilde{W}(z) = 2\left(f(pz) + \widetilde{X}(qz)\right) + \left(\widetilde{W}(pz) + \widetilde{W}(qz)\right) . \tag{12}$$

Our goal is now to solve asymptotically (as $z \to \infty$ in a cone around $\Re(z) > 0$) the above two functional equations. It is well know that equations like these are amiable to the attack of Mellin transform. To recall, for a function $f(x)$ or real valued x, we define its Mellin transform $F^*(s)$ as

$$F^*(s) = \mathcal{M}[f] = \int_0^\infty f(t)t^{s-1}dt .$$

The interested reader can find more on Mellin transform in an excellent survey by Flajolet, Gourdon and Dumas [3].

Let now

$$X(s) = \mathcal{M}[\widetilde{X}(t)] = \Gamma(s)\gamma(s) , \tag{13}$$

$$Y(s) = \mathcal{M}[\widetilde{W}(t)] = \Gamma(s)\beta(s) \tag{14}$$

be Mellin transforms of the Poisson mean and the second factorial moment. They exist in proper strips as proved in the lemma below:

Lemma 1 . *The following is true:* (i) $X(s)$ *exists for* $s \in (-b-1,-1)$*, and* $Y(s)$ *exists for* $s \in (-2b,-1)$.

(ii) *Furthermore,* $\gamma(-1-i)=0$ *for* $i=1,...,b-1$*,* $\gamma(-1-b)=(-1)^{b+1}$ *, and* $\beta(-1-i)=0$ *for* $i=1,...,b$;

Proof: Observe that $B_i(u)=i$ for $i=0,1,\ldots,b$ and thus $B_i(u)=b+(i-b)u$ for $i=1+b,\ldots,2b$ by recurrence (2). Taking derivatives, we find $\frac{\partial B_i(u)}{\partial u}=0$ for $i=0,1,\ldots,b$ and $\frac{\partial B_i(u)}{\partial u}=i-b$ for $i=b,1+b,\ldots,2b$. Furthermore, the second derivative is $\frac{\partial^2 B_i(u)}{\partial u^2}=0$ for $i=0,1,\ldots,2b$. Thus, as $z\to 0$

$$\begin{aligned}\widetilde{X}(z) &= \sum_{i=b+1}^{\infty} B_i'(1)\frac{z^i}{i!}e^{-z} = z^{(b+1)}/(b+1)! + O(z^{b+2}) \quad \text{as} \quad z\to 0 \\ \widetilde{W}(z) &= O(z^{2b+1}) \quad \text{as} \quad z \to 0\end{aligned}$$

This proves part (i), while part (ii) follows directly from the lemma below. ∎

Lemma 2 *Let* $\{x_n\}_{n=0}^{\infty}$ *be a sequence of real numbers, and let its Poisson generating function* $\widetilde{F}(z)=\sum_{i=0}^{\infty} x_n \frac{z^n}{n!}e^{-z}$ *be an entire function. Furthermore, let its Mellin transform* $F^*(s)$ *has the following factorization* $F^*(s)=\mathcal{M}[f]=\Gamma(s)\gamma(s)=\mathcal{M}[f]$*, and it exists in the strip* $(-2,-1)$ *with* $\gamma(s)$ *being analytical at* $s=-2,-3,\ldots$ *. Then,*

$$\gamma(-n)=\sum_{k=0}^{n}\binom{n}{k}(-1)^k x_k, \quad for\ n\geq 2. \tag{15}$$

Proof: Let

$$F(z)=\sum_{n=1}^{\infty} g_n\frac{z^n}{n!}, \qquad \text{and} \qquad g_n=\sum_{k=0}^{n}\binom{n}{k}(-1)^{n-k}x_k\ ,$$

and let $F_N(z)=\sum_{n=1}^{N-1} g_n\frac{z^n}{n!}$, $N>2$. Due to our assumptions, we can analytically continue $F^*(s)$ to the whole complex plane, and then for $s\in(-N,-N+1)$

$$F^*(s)=\mathcal{M}[F(z)-F_N(z)]$$

Since at $s=-N$,

$$F^*(s)=\frac{1}{s+N}\frac{(-1)^N}{N!}\gamma(-N)+O(1)\ ,$$

then by the inverse Mellin transform, we have

$$F(z)-F_N(z)=\frac{(-1)^N}{N!}\gamma(-N)z^N+O(z^{N+1}),\ as\ z\to 0\ . \tag{16}$$

But,

$$F(z)-F_N(z)=\sum_{i=N}^{\infty} g_n\frac{z^n}{n!}=g_N\frac{z^N}{N!}+O(z^{N+1})\ . \tag{17}$$

Thus, by comparing (16) and (17), we prove that $\gamma(-n)=(-1)^n g_n=\sum_{k=0}^{n}\binom{n}{k}(-1)^k x_k$ for $n>2$. ∎

Now, we are in a position to compute the Mellin transforms of $\widetilde{X}(z)$ and $\widetilde{W}(z)$. From (11) and (12), after taking Mellin transforms and using (13)-(14), we obtain

$$\begin{aligned}
\sum_{i=0}^{b}\binom{b}{i}(-1)^i\gamma(s-i) &= (p^{-s}+q^{-s})\gamma(s) \\
\sum_{i=0}^{b}\binom{b}{i}(-1)^i\beta(s-i) &= 2(p^{-s}+q^{-s})\gamma(s)+(p^{-s}+q^{-s})\beta(s)
\end{aligned}$$

and according to Lemma 1 $\gamma(s)$ exists in $\Re(s)\in(-b-1,-1)$ while $\beta(s)$ is well defined in the strip $\Re(s)\in(-2b,-1)$. To simplify it, we introduce a new notation, namely for any function $g(s)$ we define

$$\widehat{g}(s)=\sum_{i=1}^{b}\binom{b}{i}(-1)^{i+1}g(s-i) \tag{18}$$

provided $g(s-1),\ldots,g(s-b)$ exist. Then, we finally obtain

$$\gamma(s) = \frac{1}{1-p^{-s}-q^{-s}}\widehat{\gamma}(s) \tag{19}$$

$$\beta(s) = \frac{1}{1-p^{-s}-q^{-s}}\widehat{\beta}(s)+\frac{2(p^{-s}+q^{-s})}{(1-p^{-s}-q^{-s})^2}\widehat{\gamma}(s) \tag{20}$$

Let now s_k, $k=0,\pm1,\pm2,\ldots$ be roots of $1-p^{-s}-q^{-s}=0$. Observe that $s_0=-1$. At $s=s_k$ we have

$$\frac{1}{1-p^{-s}-q^{-s}}=-\frac{1}{h(s_k)}\frac{1}{s-s_k}+\frac{h_2(s_k)}{2h^2(s_k)}+O(s-s_k) \tag{21}$$

where $h(t)=-p^{-t}\log p-q^{-t}\log q$. Expending $\Gamma(s)\widehat{\gamma}(s)$ around $s=s_k$ we find

$$\Gamma(s)\widehat{\gamma}(s)=\Gamma(s_k)\widehat{\gamma}(s_k)+\left(\Gamma(s_k)\widehat{\gamma}'(s_k)+\Gamma'(s_k)\widehat{\gamma}(s_k)\right)(s-s_k)+O((s-s_k)^2)\ .$$

Observe that the main contribution to the asymptotics comes from the residue at $s=s_0=-1$. After some algebra, we find

$$\gamma(s) = -\frac{1}{h_1}\frac{1}{s+1}+\frac{h_2}{2h_1^2}-\frac{\widehat{\gamma}'(-1)}{h_1}+O(s+1) \tag{22}$$

$$\beta(s) = \frac{2}{h_1^2}\frac{1}{(s+1)^2}+\left(-2\frac{h_2-h^2}{h_1^3}+2\widehat{\gamma}'(-1)\frac{1}{h_1^2}\right)\frac{1}{s+1}+O(1)\ . \tag{23}$$

In the above, we used the fact that $\widehat{\gamma}(-1)=1$ and $\widehat{\beta}(-1)=0$, which follow directly from Lemma 2. Observe now that $\Gamma(s)=-\frac{1}{s+1}+(\gamma-1)+O(s+1)$, hence the Laurent expansion of $X(s)$ at $s=-1$ is

$$X(s)=\Gamma(s)\gamma(s)=\frac{1}{h_1}\frac{1}{(s+1)^2}-\left(\frac{h_2}{2h_1^2}-\frac{1}{h_1}\widehat{\gamma}'(-1)+\frac{\gamma-1}{h_1}\right)\frac{1}{s+1} \tag{24}$$

The above formula for $X(s)$ can be now used to obtain asymptotics of $\widetilde{X}(z)$ using standard arguments (cf. [1, 3, 9]) of the inverse Mellin transform, that is,

$$\widetilde{X}(z)=\frac{1}{2\pi i}\int_{-\frac{3}{2}-i\infty}^{-\frac{3}{2}+i\infty}X(s)z^{-s}ds\ .$$

After some algebra, we finally arrive at

$$\widetilde{X}(z) \quad = \quad \frac{1}{h_1} z \log z + \left(\frac{h_2}{2h_1^2} - \frac{1}{h_1}\widehat{\gamma}'(-1) + \frac{\gamma - 1}{h_1} \right) z + \sum_{k \neq 0} \frac{\Gamma(s_k)\widehat{\gamma}(s_k)}{h(s_k)} z^{-s_k} + O(1) \tag{25}$$

The above asymptotic formula concerns the behavior of the Poisson mean as $z \to \infty$. Our original goal was to derive asymptotics of the mean ED_m in the Bernoulli model. To infer Bernoulli model behavior from its Poisson model asymptotics, we must apply the so called *depoissonization lemma.* This lemma basically says that $mED_m \sim \widetilde{X}(m)$ under some weak conditions that are easy to verify in our case. The reader is referred to [6, 12] for more details about the depoissonization lemma. Thus,

$$ED_m = \frac{1}{h_1} \log m + \frac{h_2}{2h_1^2} - \frac{1}{h_1}\widehat{\gamma}'(-1) + \frac{\gamma - 1}{h_1} + \sum_{k \neq 0} \frac{\Gamma(s_k)\widehat{\gamma}(s_k)}{h(s_k)} m^{-1-s_k} + O\left(\frac{\log m}{\sqrt{m}} \right) .$$

Similarly, at $s = -1$,

$$Y(s) = -\frac{2}{h_1^2} \frac{1}{(s+1)^3} + \frac{2}{h_1} \left(\frac{h_2 - h_1^2}{h_1^2} - \frac{1}{h_1}\widehat{\gamma}'(-1) + \frac{\gamma - 1}{h_1} \right) \frac{1}{(s+1)^2} + O\left(\frac{1}{s+1} \right)$$

hence by the inverse Mellin transform, from the above equation we obtain

$$\widetilde{W}(z) = \frac{1}{h_1^2} z \log^2 z + \frac{2}{h_1} \left(\frac{h_2 - h_1^2}{h_1^2} - \frac{\widehat{\gamma}'(-1) + 1 - \gamma}{h_1} \right) z \log z + 2 \sum_{k \neq 0} \frac{\Gamma(s_k)\widehat{\gamma}(s_k)}{h^2(s_k)} z^{-s_k} \log z + O(1) .$$

Another application of the depoissonization lemma gives us

$$\begin{aligned} ED_m(D_m - 1) = \frac{\widetilde{W}(m)}{m} \quad &= \quad \frac{1}{h_1^2} \log^2 m + 2\frac{1}{h_1} \left(\frac{h_2 - h_1^2}{h_1^2} - \frac{1}{h_1}\widehat{\gamma}'(-1) + \frac{\gamma - 1}{h_1} \right) \log m \\ &\quad + 2 \sum_{k \neq 0} \frac{\Gamma(s_k)\widehat{\gamma}(s_k)}{h^2(s_k)} m^{-1-s_k} \log m + O\left(\frac{\log m}{\sqrt{m}} \right) . \end{aligned}$$

After some manipulation, we finally arrive at

$$\begin{aligned} \text{Var } D_m \quad &= \quad \widetilde{W}(m)/m + ED_m - (ED_m)^2 \\ &= \quad \frac{h_2 - h_1^2}{h_1^3} \log m + 2 \sum_{k \neq 0} \frac{\Gamma(s_k)\widehat{\gamma}(s_k)}{h(s_k)} \left(\frac{1}{h(s_k)} - \frac{1}{h_1} \right) m^{-1-s_k} \log m + C \end{aligned} \tag{26}$$

where C is a constant. If $\Re(s_k) = -1$, then $h(s_k) = h_1$, and

$$\text{Var } D_m = \frac{h_2 - h_1^2}{h_1^3} \log m + O(1) .$$

If $\Re(s_k) > -1$, then $m^{-1-s_k} \log m = o(1)$, thus $\text{Var } D_m = \frac{h_2 - h_1^2}{h_1^3} \log m + C$. We know that $\Re(s_k) = -1$ whenever $\log p / \log q$ is rational (cf. [5]), otherwise $\Re(s_k) > -1$.

Finally, we must show how to evaluate the constant $\widehat{\gamma}'(-1)$. Observe that $\gamma(s) = \mathcal{M}[\widetilde{X}(z)]$ and $\widetilde{X}(z) = \sum_{n=1}^{\infty} f_n \frac{z^n}{n!} e^{-z}$ where the sequence $f_n = ED_n$ is defined in Theorem 1(i). Thus, since $\mathcal{M}[z^n e^{-z}] = \Gamma(s + n)$, we obtain

$$\gamma(s) = \sum_{n=b+1}^{\infty} \frac{f_n}{n!} s(s+1)...(s+n-1)$$

for $\Re(s) \in (-b-2,-1)$. Hence,

$$\gamma'(s) = \sum_{n=b+1}^{\infty} \frac{f_n}{n!} s(s+1)...(s+n-1) \sum_{i=0}^{n-1} \frac{1}{s+i} .$$

After some tedious manipulation, one can evaluate $\widetilde{\gamma}'(-1)$ as in (5). Thus, Theorem 1(i) is proved.

References

[1] P. Flajolet and R. Sedgewick, Digital Search Trees Revisited, *SIAM J. Computing*, 15, 748-767 (1986).

[2] P. Flajolet and B. Richmond, Generalized Digital Trees and Their Difference-Differential Equations, *Random Structures & Algorithms*, 3, 305-320 (1992).

[3] P. Flajolet, X. Gourdon, P. Dumas, Mellin transforms and asymptotics: harmonic sums, *Theoretical Computer Science*, 144, 3-58, 1995.

[4] F. Hubalek, *Beiträge zur Analyse Verallgemeinerter Digitaler Suchbäume*, Ph. D. Technische Universität Wien, 1994.

[5] P. Jacquet and W. Szpankowski, Asymptotic Behavior of the Lempel-Ziv Parsing Scheme and Digital Search Trees, *Theoretical Computer Science*, June 1995.

[6] P. Jacquet and W. Szpankowski, Analytical Depoissonization Lemma and Its Applications, preprint.

[7] D. Knuth, *The Art of Computer Programming. Sorting and Searching,* Addison-Wesley (1973).

[8] G. Louchard, Exact and Asymptotic Distributions in Digital and Binary Search Trees, *RAIRO Theoretical Inform. Applications*, 21, 479-495 (1987).

[9] G. Louchard and W. Szpankowski, Average Profile and Limiting Distribution for a Phrase Size in the Lempel-Ziv Parsing Algorithm, *IEEE Information Theory*, 41, 478-488, 1995.

[10] G. Louchard and W. Szpankowski, Generalized Lempel-Ziv Parsing Scheme and its Preliminary Analysis of the Average Profile, *Proc. Data Compression Conference*, 262-271, Snowbird, 1995.

[11] H. Mahmoud, *Evolution of Random Search Trees*, John Wiley & Sons, New York (1992).

[12] B. Rais, P. Jacquet and W. Szpankowski, A Limiting Distribution for the Depth in PATRICIA Tries, *SIAM J. Discrete Mathematics*, 6, 197-213 (1993).

[13] W. Szpankowski, The Evaluation of an Alternating Sum with Applications to the Analysis of Some Data Structures, *Information Processing Letters*, 28, 13-19 (1988).

[14] W. Szpankowski, A Characterization of Digital search Trees From the Successful Search Viewpoint, *Theoretical Computer Science*, 85, 117-134 (1991).

[15] A. Wyner and J. Ziv, Some Asymptotic Properties of the Entropy of a Stationary Ergodic Data Source with Applications to Data Compression, *IEEE Trans. Information Theory*, 35, 1250-1258 (1989).

[16] J. Ziv and A. Lempel, Compression of Individual Sequences via Variable-rate Coding, *IEEE Trans. Information Theory*, 24, 530-536 (1978).

The Redundancy of Source Coding with a Fidelity Criterion*

Zhen Zhang[1], En-hui Yang[2], and Victor K. Wei[3]

Abstract — **The problem of redundancy of source coding with respect to a fidelity criterion is considered. For any fixed rate $R > 0$ and any memoryless source with finite source and reproduction alphabets and a common distribution p, the n-th order distortion redundancy $\mathcal{D}_n(R)$ of fixed-rate coding is defined as the minimum of the difference between the expected distortion per symbol of any block code with length n and rate R and the distortion rate function $d(p, R)$ of the source p. It is demonstrated that for sufficiently large n, $\mathcal{D}_n(R)$ is equal to $-\frac{\partial}{\partial R}d(p, R)\ln n/2n + o(\ln n/n)$, where $\frac{\partial}{\partial R}d(p, R)$ is the partial derivative of $d(p, R)$ evaluated at R and assumed to exist. For any fixed distortion level $d > 0$ and any memoryless source p, the n-th order rate redundancy $\mathcal{R}_n(d)$ of coding at fixed distortion level d (or by using d-semifaithful codes) is defined as the minimum of the difference between the expected rate per symbol of any d-semifaithful code of length n and the rate distortion function $R(p, d)$ of p evaluated at d. It is proved that for sufficiently large n, $\mathcal{R}_n(d)$ is upper bounded by $\ln n/n + o(\ln n/n)$ and lower bounded by $\ln n/2n + o(\ln n/n)$. As a by-product, the lower bound of $\mathcal{R}_n(d)$ derived in this paper gives a positive answer to a recent conjecture proposed by Yu and Speed.**

I. INTRODUCTION

The rate distortion function gives the minimum number of bits per symbol needed to convey in order to reproduce source with an average distortion that does not exceed a specific threshold. For any d-semifaithful code, the performance of the code can be measured by the rate redundancy of the code which is defined to be the difference between the expected codeword length per symbol of the code and the rate distortion function. The less redundancy, the better the d-semifaithful code. For any block code with rate R, the performance of the code can be measured by the distortion redundancy of the code which is defined to be the difference between the expected distortion per symbol of the code and the distortion rate function. Pilc[3] considered for the first time the problem of distortion redundancy. For memoryless sources with finite source and reproduction alphabets and fixed-rate coding, he proved that the n-th order distortion redundancy of fixed-rate coding of a memoryless source p is upper bounded by $(-(1+\epsilon)\frac{\partial}{\partial R}d(p, R)\ln n/2n)(1 + o(1))$ and argued that the n-th order distortion redundancy is lower bounded by $(-\frac{\partial}{\partial R}d(p, R)\ln n/2n)(1 + o(1))$, where $\frac{\partial}{\partial R}d(p, R)$ is the derivative of the distortion rate function $d(p, R)$ with respect to R and n is the blocklength. Recently, Yu and Speed[4] proved that for memoryless sources with finite source and reproduction alphabets and d-semifaithful codes, the n-th order universal rate redundancy of coding p by using d-semifaithful codes is upper bounded by $(KJ + J + 4)\ln n/n + o(n^{-1})$ and conjectured that $O(\ln n/n)$ is the optimal rate at which the n-th order rate redundancy converges to 0 as $n \to \infty$, where J and K are the sizes of the source alphabet and the reproduction

---

*This work was supported in part by National Science Foundation under grants NCR-9205265 and NCR-9508282.

[1] Zhen Zhang is with Communication Sciences Institute, Dept. of Electrical Engineering- Systems, University of Southern California, Los Angeles, CA 90089-2565. Email: zzhang@milly.usc.edu

[2] En-hui Yang is with the Dept. of Mathematics, Nankai University, Tianjin 300071, P.R. China. Email: ehyang@milly.usc.edu

[3] Victor K. Wei is with the Dept. of Information Engineering, the Chinese University of Hong Kong, Shatin, New Territory, Hong Kong.

alphabet, respectively. Linder, Lugosi and Zeger[2] considered the case of real alphabet and proved that for memoryless sources, the n-th order universal distortion redundancy is upper bounded by $O(\sqrt{\ln n/n})$. Unfortunately, Pilc's lower bound argument is heavily based upon an unjustified assumption and therefore is incomplete. Whether or not the Pilc's lower bound is true was still an open problem until the present paper, where we resolve the matter.

The aim of this paper is to investigate the distortion redundancy by using fixed-rate codes and the rate redundancy by using d-semifaithful codes for memoryless sources with finite source and reproduction alphabets. For distortion redundancy, we develop a new approach to the lower-bound-proving and prove that the lower bound Pilc claimed turns out to be true. For rate redundancy, we prove that for memoryless source p and sufficiently large n, the n-th order rate redundancy is upper bounded by $\ln n/n + o(\ln n/n)$ and lower bounded by $\ln n/2n + o(\ln n/n)$. As a byproduct, the lower bound on the n-th order rate redundancy we obtain gives a positive answer to a recent conjecture of Yu and Speed [4]. During the process of investigating distortion redundancy and rate redundancy, we develop a deep theory on types and d-ball covering.

II. STATEMENT OF MAIN RESULTS

Let $\mathbf{A}$ and $\mathbf{B}$ be two nonempty finite alphabets. $\mathbf{A}$ and $\mathbf{B}$ will serve as our source alphabet and reproduction alphabet, respectively. Without loss of generality, we shall assume $\mathbf{A} = \{1, \cdots, J\}$ and $\mathbf{B} = \{1, \cdots, K\}$. Let $\{X_t\}_{t=1}^{\infty}$ be an *i.i.d* source with finite alphabet $\mathbf{A}$. Let p be the generic probability distribution of the source $\{X_t\}_{t=1}^{\infty}$, that is, $\Pr(X_t = j) = p_j$ for any $j \in \mathbf{A}$ and $t > 0$. Let $\rho : \mathbf{A} \times \mathbf{B} \to [0, \infty)$ be a single letter distortion measure satisfying the condition $\max_{x\in\mathbf{A}} \min_{y\in\mathbf{B}} \rho(x,y) = 0$. Denote by $R(p,d)$ the rate distortion function of the source $\{X_t\}_1^{\infty}$ with respect to the single letter fidelity criterion $\{\rho_n\}$ generated by ρ, where $\rho_n : \mathbf{A}^n \times \mathbf{B}^n \to [0,\infty)$ is the map in which $\rho_n(x^n, y^n) = n^{-1}\sum_{i=1}^{n} \rho(x_i, y_i)$ for any $x^n \in \mathbf{A}^n$ and y^n. From rate distortion theory [1], it is well-known that $R(p,d)$ is given by the following minimization problem:

$$R(p,d) = \inf_{Q\in\mathcal{Q}_d} I(p;Q)\,, \qquad (2.1)$$

where $I(p;Q) = \sum_{k,j} p_j Q_{k|j} \ln \frac{Q_{k|j}}{Q_k}$, $(Q_{k|j})_{J\times K}$ is a matrix of transition probabilities, $Q_k = \sum_j p_j Q_{k|j}$ and

$$\mathcal{Q}_d = \{(Q_{k|j})_{J\times K} : \sum_{j,k} p_j Q_{k|j}\rho(j,k) \le d\}\,. \qquad (2.2)$$

The inverse of $R(p,d)$ is the distortion rate function of the source $\{X_t\}_1^{\infty}$ which is denoted by $d(p,R)$. Throughout the paper, the rate distortion function $R(p,d)$ and the coding rates of block codes and d-semifaithful codes are measured in terms of nats.

Let $(Q^*_{k|j})_{J\times K}$ be an optimal transition probability matrix which achieves the minimum in (2.1). For each $k \in \mathbf{B}$, let $Q^*_k = \sum_{j=1}^{J} p_j Q^*_{k|j}$. Q^* will be called an optimal distribution on $\mathbf{B}$ associated with p and d. From [1], we have the following facts: If $Q^*_{k|j} = 0$ for some $j \in \mathbf{A}$, then $Q^*_k = 0$; $Q^*_{k|j} = Q^*_k e^{s\rho(j,k)} (\sum_{k'} Q^*_{k'} e^{s\rho(j,k')})^{-1}$ where s is the slope of $R(p,d)$ at point d whenever $0 < d < d_{max} \overset{def}{=} \min_{y\in\mathbf{B}} \sum_{x\in\mathbf{A}} p(x)\rho(x,y)$. Without loss of generality, we shall assume that $Q^*_k > 0$ for any $k \in \mathbf{B}$. Otherwise some letters can be deleted from $\mathbf{B}$. Throughout the paper, for simplicity, the optimal distribution Q^* on $\mathbf{B}$ shall be assumed to be unique.

We employ two kinds of codes: block codes and d-semifaithful codes. A block code $\mathbf{C}_n$ of order n is simply a subset of $\mathbf{B}^n$. The block code $\mathbf{C}_n$ of order n is said to operate at the rate level R if $|\mathbf{C}_n| \le e^{nR}$. (Here and henceforth, $|S|$ denotes the cardinality of S if S is a finite set.) When the block code $\mathbf{C}_n$ is used to encode the source $\{X_t\}_1^{\infty}$, the resulting average distortion is defined to be $\rho_n(\mathbf{C}_n) = \mathbf{E}\rho_n(X^n, \mathbf{C}_n)$ where for each $x^n \in \mathbf{A}^n$, $\rho_n(x^n, \mathbf{C}_n) = \min_{c\in\mathbf{C}_n} \rho_n(x^n, c)$. The performance of the block code $\mathbf{C}_n$ is measured by the distortion redundancy of the code which is defined to be $D_n(\mathbf{C}_n) = \rho_n(\mathbf{C}_n) - d(p,R)$.

From rate distortion theory[1], it follows that $D_n(\mathbf{C}_n)$ is nonnegative. Define the n-th order distortion redundancy of fixed-rate coding of the source p as follows

$$\mathcal{D}_n(R) = \min\{D_n(\mathbf{C}_n)\} , \qquad (2.3)$$

where the minimum is taken over all block codes $\mathbf{C}_n$ of order n operating at rate level R. One of the goals of this paper is to determine for sufficiently large n the n-th order distortion redundancy $\mathcal{D}_n(R)$.

Theorem 1 *Let $R > 0$. Assume the optimal distribution Q^* on $\mathbf{B}$ associated with p and $d(p, R)$ is unique. Then for sufficiently large n,*

$$\mathcal{D}_n(R) = -\frac{\partial}{\partial R} d(p, R) \frac{\ln n}{2n} + o\left(\frac{\ln n}{n}\right) , \quad (2.4)$$

where $\partial d(p, R)/\partial R$ is the derivative of $d(p, R)$ with respect to R.

Remark 1 Pilc[3] proved that for any $\epsilon > 0$

$$\mathcal{D}_n(R) \le -(1+\epsilon) \frac{\partial}{\partial R} d(p, R) \frac{\ln n}{2n} + o(\frac{\ln n}{n}) ,$$

for sufficiently large n. He claimed that

$$\mathcal{D}_n(R) \ge -\frac{\partial}{\partial R} d(p, R) \frac{\ln n}{2n} + o\left(\frac{\ln n}{n}\right) .$$

Unfortunately, his argument to the lower bound turns out to be incomplete.

Theorem 1 will be proved via the following two theorems.

Theorem 2 *Let $R > 0$. Assume the optimal distribution Q^* on $\mathbf{B}$ associated with p and $d(p, R)$ is unique. Then for sufficiently large n and any block code $\mathbf{C}_n$ of order n operating at rate level R,*

$$\rho_n(\mathbf{C}_n) \ge d(p, R) - \frac{\partial}{\partial R} d(p, R) \frac{\ln n}{2n} + o\left(\frac{\ln n}{n}\right) . \quad (2.5)$$

Remark 2 It should be noted that the lower bound in (2.5) does not include the case of $R = 0$. In fact, one can prove that the distortion redundancy $\mathcal{D}_n(0)$ is actually equal to 0.

Theorem 3 *There exists for sufficiently large n a block code $\mathbf{C}_n$ of order n operating at rate level R such that*

$$\rho_n(\mathbf{C}_n) \le d(p, R) - \frac{\partial}{\partial R} d(p, R) \frac{\ln n}{2n} + o\left(\frac{\ln n}{n}\right) . \quad (2.6)$$

A d-semifaithful code $\mathcal{C}_n$ of order n is a triple $\mathcal{C}_n = (\phi_n, G_n, \psi_n)$, where G_n is a subset of $\mathbf{B}^n$, ϕ_n is a map from $\mathbf{A}^n$ to G_n satisfying $\rho_n(x^n, \phi_n(x^n)) \le d$ for each $x^n \in \mathbf{A}^n$, and ψ_n is a one to one map from G_n to a set of finite binary words satisfying the prefix condition. When the d-semifaithful code $\mathcal{C}_n$ is used to encode the source p, the resulting average rate per symbol is defined to be

$$r_n(\mathcal{C}_n) = \mathbf{E}(\text{length of } \psi_n(\phi_n(X^n))/n) .$$

The performance of the d-semifaithful code $\mathcal{C}_n$ is measured by the rate redundancy which is defined to be

$$R_n(\mathcal{C}_n) = r_n(\mathcal{C}_n) - R(p, d) .$$

From rate distortion theory, it is well-known that $R_n(\mathcal{C}_n)$ is nonnegative. Define the n-th order rate redundancy of coding of the source p by using d-semifaithful codes as follows

$$\mathcal{R}_n(d) = \min\{R_n(\mathcal{C}_n)\} , \qquad (2.7)$$

where the minimum is taken over all d-semifaithful codes $\mathcal{C}_n$ of order n.

We also assume $R(p, d) > 0$. The following two theorems give a lower bound and an upper bound to $\mathcal{R}_n(d)$, respectively.

Theorem 4 *Let Q^* be the unique optimal distribution on $\mathbf{B}$ associated with p and d, then for sufficiently large n and any d-semifaithful code $\mathcal{C}_n$ of order n,*

$$r_n(\mathcal{C}_n) \ge R(p, d) + \frac{\ln n}{2n} + o\left(\frac{\ln n}{n}\right) . \quad (2.8)$$

It is easy to see that Theorem 4 gives a positive answer to the previously mentioned conjecture of Yu and Speed [4].

Remark 3 The lower bound in (2.8) does not include the case of $R(p,d)=0$. In fact, $\mathcal{R}_n(d)$ may equal 0 in the case of $R(p,d)=0$.

Theorem 5 *There exists for sufficiently large n a d-semifaithful code $\mathcal{C}_n$ of order n such that*

$$r_n(\mathcal{C}_n) \le R(p,d) + \frac{\ln n}{n} + o\left(\frac{\ln n}{n}\right) . \quad (2.9)$$

III. TYPES AND d-BALL COVERING

We develop a theory on types and d-ball covering, which will play a key role in the proofs of the theorems. First of all, we review some basic facts on types and d-ball covering. Let $\mathbf{M} = \{e_1, \cdots, e_m\}$ be a finite set and let $\mathcal{P}(\mathbf{M})$ be the set of all probability distributions on $\mathbf{M}$. A $t \in \mathcal{P}(\mathbf{M})$ is said to be an n-type if for any $e \in \mathbf{M}$, $t(e) \in \{0, \frac{1}{n}, \frac{2}{n}, \cdots, 1\}$. The set of all n-types on $\mathbf{M}$ is denoted by $\mathcal{T}_n(\mathbf{M})$. We have $|\mathcal{T}_n(\mathbf{M})| = \binom{n+m-1}{m-1}$. The *type* of a word $z^n \in \mathbf{M}^n$ is defined as $t(z^n) = (t(z^n, e_1), \cdots, t(z^n, e_m))$ which is an n-type on $\mathbf{M}$, where $t(z^n, e_i) = \frac{1}{n}|\{j : z_j = e_i\}|$. For $t \in \mathcal{T}_n(\mathbf{M})$, define $T^n_{\mathbf{M}}(t) = \{z^n \in \mathbf{M}^n : t(z^n) = t\}$. In this paper, we are interested in three alphabets: the source alphabet $\mathbf{A}$, the reproduction alphabet $\mathbf{B}$, and the joint alphabet $\mathbf{A} \times \mathbf{B}$. Let s be a joint n-type in $\mathcal{T}_n(\mathbf{A} \times \mathbf{B})$, then $t(j) = \sum_{k \in \mathbf{B}} s(j,k)$ defines an n-type in $\mathcal{T}_n(\mathbf{A})$ which is called the marginal n-type of s on $\mathbf{A}$. Similarly, we can define the marginal n-type r of s on $\mathbf{B}$. For these three alphabets, we use $T^n_X(t)$, $T^n_Y(r)$ and $T^n_{X,Y}(s)$ to denote $T^n_{\mathbf{A}}(t)$, $T^n_{\mathbf{B}}(r)$ and $T^n_{\mathbf{A}\times\mathbf{B}}(s)$, respectively.

We now introduce some basic estimates for the cardinalities and probabilities of these sets.

Fact 1 *If $t(j) > 0$ for each $j \in \mathbf{A}$, then*

$$|T^n_X(t)| = e^{O(1) - \frac{J-1}{2}\ln n + nH(t)} , \quad (3.1)$$

where $H(t)$ is the entropy of the distribution t in terms of nats. Similar results hold also for $T^n_Y(r)$ and $T^n_{X,Y}(s)$ for type r satisfying $r(k) > 0$ for each $k \in \mathbf{B}$ and type s satisfying $s(j,k) > 0$ for $(j,k) \in \mathbf{A} \times \mathbf{B}$.

Fact 2 *Under the same conditions as in Fact 1, we have*

$$p^n(T^n_X(t)) = e^{O(1) - \frac{J-1}{2}\ln n - nI(t\|p)} . \quad (3.2)$$

Similar results hold for $Q^n(T^n_Y(r))$ and $W^n(T^n_{X,Y}(s))$ where $W^n(x^n, y^n) = p^n(x^n)Q^n(y^n|x^n)$, $Q^n(y^n|x^n) = \Pi_{i=1}^n Q(y_i|x_i)$, $Q^n(y^n) = \sum_{x^n} W^n(x^n, y^n)$ and $I(\mu_1||\mu_2)$ denotes the relative entropy between two distributions μ_1 and μ_2.

For any $y^n \in \mathbf{B}^n$, let $B(y^n, d) = \{x^n \in \mathbf{A}^n : \rho_n(x^n, y^n) \le d\}$. The set $B(y^n, d)$ can be conceptualized as a ball centered at y^n with radius d. For $t \in \mathcal{T}_n(\mathbf{A})$, define the restricted d-ball as $B(y^n, t, d) = B(y^n, d) \cap T^n_X(t)$. It is easy to see that the cardinality of $B(y^n, t, d)$ depends on y^n only through the type $t(y^n)$ of y^n. Therefore, we can define a function $F_n(r, t, d)$ by $F_n(r,t,d) = |B(y^n, t, d)|$, where r is an n-type on $\mathbf{B}$ and $y^n \in T^n_Y(r)$.

Let $\mathcal{S}_n(r,t,d)$ denote the set consisting of all $s \in \mathcal{T}_n(\mathbf{A} \times \mathbf{B})$ such that r and t are the two marginals of s and $E_s\rho(X,Y) \le d$, where $E_s\rho(X,Y) = \sum_{j\in\mathbf{A},k\in\mathbf{B}} s(j,k)\rho(j,k)$. A simple counting argument shows that

$$F_n(r,t,d) =$$

$$\sum_{s \in \mathcal{S}_n(r,t,d)} \prod_{k \in \mathbf{B}} \binom{nr(k)}{ns(1,k), ns(2,k), \cdots, ns(J,k)} .$$

Letting

$$H(s|r) = \sum_{j\in\mathbf{A},k\in\mathbf{B}} -s(j,k)\ln s(j,k) + \sum_{k\in\mathbf{B}} r(k)\ln r(k)$$

be the conditional entropy of t given r through the joint distribution s, we have

$$F_n(r,t,d) = \exp\{o(n) + n \max_{s\in\mathcal{S}_n(r,t,d)} H(s|r)\}$$

$$= \exp\{o(n) + n \max_{s\in\mathcal{S}_n(r,t,d)} H(s) - nH(r)\} \quad (3.3)$$

To study the function $\max_{s\in\mathcal{S}_n(r,t,d)} H(s)$, we now drop the requirement that s, t and r be

n-types and regard them as arbitrary probability distributions. This leads us to define a so-called upper joint entropy $H_u(r,t,d)$ by

$$H_u(r,t,d) = \sup_{s\in\mathcal{S}(r,t,d)} H(s) \,, \tag{3.4}$$

where $\mathcal{S}(r,t,d)$ denotes the set consisting of all $s \in \mathcal{P}(\mathbf{A}\times\mathcal{B})$ such that r and t are the two marginals of s and $E_s\rho(X,Y) \le d$, $H(s)$ is the entropy of s, and r and t are distributions of $\mathcal{P}(\mathbf{B})$ and $\mathcal{P}(\mathbf{A})$, respectively. Using the upper joint entropy $H_u(r,t,d)$ defined above, we can now rewrite (3.3) as follows

$$F_n(r,t,d) = e^{o(n)+nH_u(r,t,d)-nH(r)} \,. \tag{3.5}$$

(Careful readers may find that there is a gap between $\max_{s\in\mathcal{S}_n(r,t,d)} H(s)$ and $H_u(r,t,d)$ when r and t are n-types. But actually this gap is $O(1/n)$.) For any $y^n \in T^n_Y(r)$,

$$\begin{aligned} & p^n(B(y^n,t,d)) \\ = \; & F(r,t,d)\exp\{n\sum_{j\in\mathbf{A}} t(j)\ln p_j\} \\ = \; & \exp\{o(n) + n(-I(t\|p) - H(t) \\ & - H(r) + H_u(r,t,d))\} \,. \end{aligned} \tag{3.6}$$

Letting

$$I_l(r,t,d) = H(r) + H(t) - H_u(r,t,d) \,, \tag{3.7}$$

which is called the lower mutual information between distributions r and t with respect to the distortion d, we can rewrite (3.6) as

$$\begin{aligned} & p^n(B(y^n,t,d)) = \\ & \exp\{o(n) - n(I(t\|p) + I_l(r,t,d))\} \,. \end{aligned} \tag{3.8}$$

The reason why we introduce the upper joint entropy $H_u(r,t,d)$ and the lower mutual information $I_l(r,t,d)$ is that we want to obtain more accurate estimates for $F_n(r,t,d)$ and $p^n(B(y^n,t,d))$ than those given in (3.5) and (3.8), respectively. To do this, we first need to analyze the properties of the upper joint entropy $H_u(r,t,d)$ and the lower mutual information $I_l(r,t,d)$.

Note that when d is small enough, $\mathcal{S}(r,t,d)$ may be empty. Therefore, we define $d_{min}(r,t) = \min\{d : |\mathcal{S}(r,t,d)| > 0\}$. On the other hand, when d is large enough, $\mathcal{S}(r,t,d)$ may consists of all joint distributions s on $\mathbf{A}\times\mathbf{B}$ so that s has a marginal r on $\mathbf{B}$ and a marginal t on $\mathbf{A}$ and hence $H_u(r,t,d)$ is equal to a constant $H(t)+H(r)$. Therefore, it is convenient to define $d_{max}(r,t) = \sup\{d | H_u(r,t,d) < H(t) + H(r)\}$. We have the following results

Lemma 1 *The functions $H_u(r,t,d)$ and $I_l(r,t,d)$ have the following properties:*

1. *$H_u(r,t,d)$ is a concave function of r,t,d and for any fixed t, $I_l(r,t,d)$ as a function of r and d is convex;*
2. *$H_u(r,t,d)$ is uniformly upper semi-continuous in its domain and $I_l(r,t,d)$ is uniformly lower semi-continuous in its domain;*
3. *for any fixed r,t, $H_u(r,t,d)$ is a strictly increasing function in $[d_{min}(r,t), d_{max}(r,t)]$;*
4. *$R(t,d) = \inf\{I_l(r,t,d) | r \in \mathcal{P}(\mathbf{B})\}$, where $R(t,d)$ is the rate distortion function of the distribution t.*

Lemma 2 *For any point (r_0,t_0,d_0), where $r_0 \in \mathcal{P}(\mathbf{B})$ satisfies $r_0(k) > 0$ for any $k \in \mathbf{B}$, $t_0 \in \mathcal{P}(\mathbf{A})$ satisfies $t_0(j) > 0$ for any $j \in \mathbf{A}$, and $d_0 \in (d_{min}(r_0,t_0), d_{max}(r_0,t_0))$, there exists a small neighborhood $\mathcal{N}(r_0,t_0,d_0)$ of (r_0,t_0,d_0), in which*

$$\begin{aligned} & \mathcal{N}(r_0,t_0,d_0) = \{(r,t,d) | r\in\mathcal{P}(\mathbf{B}), t\in\mathcal{P}(\mathbf{A}), \\ & \|r-r_0\| \le \sigma, \|t-t_0\| \le \sigma, \& |d-d_0| \le \sigma\} \end{aligned} \tag{3.9}$$

for $\sigma > 0$, such that

(1) *$H_u(r,t,d)$ is at least second order differentiable in $\mathcal{N}(r_0,t_0,d_0)$;*

(2) *$\partial H_u(r,t,d)/\partial d > 0$;*

(3) *$\partial H_u(r,t,d)/\partial d$ is bounded in $\mathcal{N}(r_0,t_0,d_0)$, that is, there exist two positive real numbers b_1 and b_2 such that for any $(r,t,d) \in \mathcal{N}(r_0,t_0,d_0)$*

$$b_1 \le \frac{\partial}{\partial d} H_u(r,t,d) \le b_2 \,. \tag{3.10}$$

In the following discussion, σ in (3.9) is assumed to be selected small enough to guarantee that there exists a $\sigma_1 > 0$ such that for any $(r, t, d) \in \mathcal{N}(r_0, t_0, d_0)$, $r(k) \geq \sigma_1$ and $t(j) \geq \sigma_1$ for all j and k. The next lemma is crucial in the proofs of our main results on distortion redundancy and rate redundancy.

Lemma 3 *For sufficiently large n and for any $(r, t, d) \in \mathcal{N}(r_0, t_0, d_0)$, where r and t are n-types,*

$$\ln F_n(r, t, d) \leq$$

$$nH_u(r, t, d) - nH(r) - \frac{J}{2} \ln n + c_1 \ln \ln n \quad (3.11)$$

and

$$\ln F_n(r, t, d) \geq$$

$$nH_u(r, t, d) - nH(r) - \frac{J}{2} \ln n - c_2 \, , \quad (3.12)$$

where c_1 and c_2 are positive constants depending only on $\mathcal{N}(r_0, t_0, d_0)$.

As a consequence of Lemma 3, we have

Lemma 4 *For sufficiently large n and any $(r, t, d) \in \mathcal{N}(r_0, t_0, d_0)$, where r and t are n-types,*

$$-\ln p^n(B_n(r, t, d)) =$$

$$nI(t||p) + nI_l(r, t, d) + \frac{J}{2} \ln n + o(\ln n) \, .$$

For any $y^n \in \mathbf{B}^n$ and $t \in \mathcal{T}_n(\mathbf{A})$, it is not hard to see that the following quantity

$$|B_n(y^n, t, d)|^{-1} \sum_{x^n \in B_n(y^n, t, d)} \rho_n(x^n, y^n)$$

depends on y^n only through the type $t(y^n)$ of y^n. Therefore we can define a function $\rho_n(r, t, d)$ by

$$\rho_n(r, t, d) =$$

$$|B_n(y^n, t, d)|^{-1} \sum_{x^n \in B_n(y^n, t, d)} \rho_n(x^n, y^n) \, , \quad (3.13)$$

where y^n is any sequence of $\mathbf{B}^n$ with type $t(y^n) = r$. Let $\mathcal{N}'(r_0, t_0, d_0)$ be a subset of $\mathcal{N}(r_0, t_0, d_0)$ which is defined as

$$\mathcal{N}'(r_0, t_0, d_0) = \{(r, t, d) | r \in \mathcal{P}(\mathbf{B}), t \in \mathcal{P}(\mathbf{A}),$$

$$||r - r_0|| \leq \sigma', ||t - t_0|| \leq \sigma' \& |d - d_0| \leq \sigma'\} \, ,$$

where $0 < \sigma' < \sigma$. Using Lemma 3, we can prove the following lemma.

Lemma 5 *For sufficiently large n and any $(r, t, d) \in \mathcal{N}'(r_0, t_0, d_0)$, where r and t are n-types,*

$$\rho_n(r, t, d) = d - o(\frac{\ln n}{n}) \, . \quad (3.14)$$

IV. DISCUSSION

Because of the length limitation imposed on the paper, there is no much space for us to give the proofs of the above results. The reader is refered to [5] for proofs of all lemmas and theorems. In this section, instead, we give a heuristic argument for Theorem 2. We believe that it will be helpful in understanding the basic ideas used in the proof of this key result. Let $\mathbf{C}_n$ be a block code of order n operating at rate level R. We need to consider only the case where $\mathbf{C}_n$ is "good" in the sense that the average distortion $\rho_n(\mathbf{C}_n)$ is close enough to $d(p, R)$ since otherwise nothing needs to be proved. In this case, it can be shown that almost all codewords $c \in \mathbf{C}_n$ are "good" in the sense that their n-types $t(c)$ are close enough to Q^*, and that the contribution of "bad" codewords $c \in \mathbf{C}_n$ to the average distortion $\rho_n(\mathbf{C}_n)$ is negligible. Let $t \in \mathcal{T}_n(\mathbf{A})$ be a typical n-type which is close enough to p. On average, the cardinality of the intersection of the Voronoi cell of a good codeword $c \in \mathbf{C}_n$ with $T_X^n(t)$ is

$$\frac{\exp\{o(\ln n) - \frac{J-1}{2} \ln n + nH(p)\}}{|\mathbf{C}_n|} =$$

$$\exp\{o(\ln n) - \frac{J-1}{2} \ln n + n(H(p) - R)\} \, . \quad (4.1)$$

To get a lower bound on $\rho_n(\mathbf{C}_n)$, we form the largest restricted d-ball $B_n(c, t, d)$ centered at c so that

$$|B_n(c, t, d)| \leq$$

$$\exp\{o(\ln n) - \frac{J-1}{2} \ln n + n(H(p) - R)\} \, .$$

Denote the largest restricted d-ball $B_n(c,t,d)$ by $B_n(c,t,d_{\max})$. Clearly, the average distortion over the intersection of the Voronoi cell of the good codeword c with $T_X^n(t)$ is lower bounded by the average distortion over $B_n(c,t,d_{\max})$; by Lemma 5, the latter is further lower bounded by $d_{\max} - o(\ln n/n)$. Since $c \in \mathbf{C}_n$ is a good codeword, it follows from Lemma 3 that on average,

$$\begin{aligned} &|B_n(c,t,d_{\max})| \\ &\leq \exp\{O(\ln\ln n) - \frac{J}{2}\ln n \\ &\quad - nI_l(Q^*,p,d_{\max}) + nH(p)\} \\ &\leq \exp\{O(\ln\ln n) - \frac{J}{2}\ln n \\ &\quad - nR(p,d_{\max}) + nH(p)\} . \qquad (4.2) \end{aligned}$$

By equating the term on the right hand side of (4.2) with the right hand side term of (4.1), we get a lower bound d of $d_{\max}$, where d is the solution to the following equation

$$\begin{aligned} &O(\ln\ln n) - \frac{J}{2}\ln n - nR(p,d) + nH(p) \\ &= o(\ln n) - \frac{J-1}{2}\ln n + n(H(p) - R) \end{aligned}$$

which is equivalent to

$$R(p,d) = R - \frac{1}{2}\frac{\ln n}{n} + o(\frac{\ln n}{n}) .$$

From this, we have

$$\begin{aligned} d_{\max} &\geq d(p, R - \frac{1}{2}\frac{\ln n}{n} + o(\frac{\ln n}{n})) \\ &= d(p,R) - \frac{\partial}{\partial R}d(p,R)\frac{\ln n}{2n} + o(\frac{\ln n}{n}) \end{aligned}$$

which implies that on average

$$\rho_n(\mathbf{C}_n) \geq d(p,R) - \frac{\partial}{\partial R}d(p,R)\frac{\ln n}{2n} + o(\frac{\ln n}{n}) .$$

To conclude the paper, we list a few key open problems in this line of research:

1. Generalize the main results of this paper to memoryless sources with continuous alphabets.
2. Investigate the distortion-redundancy of fixed rate universal codes and the rate-redundancy of universal d-semifaithful codes for memoryless sources with finite source and reproduction alphabets.
3. Determine the rate of convergence of optimal fixed rate universal codes for sources with continuous alphabet. Is the result of Linder, Lugosi and Zeger[2] tight?
4. Instead of studying the trade-off between redundancy and block length of codes, investigate the trade-off between redundancy and other important parameters of coding schemes such as delay, encoding complexity, etc.

REFERENCES

[1] T. Berger, *Rate Distortion Theory.* Englewood Cliffs, NJ: Prentice-Hall, 1971.

[2] T. Linder, G. Lugosi, and K. Zeger, "Rates of convergence in the source coding theorem, in empirical quantizer design, and in universal lossy source coding," to appear in *IEEE Trans. Inform. Theory for publication*, Nov. 1994.

[3] R. J. Pilc, "The transmission distortion of a source as a function of the encoding block length," *Bell Syst. Tech. J.*, Vol. 47, pp. 827–885, 1968.

[4] B. Yu and T. P. Speed, "A rate of convergence result for a universal D-semifaithful code," *IEEE Trans. Inform. Theory*, Vol. IT-39, pp. 813–820, 1993.

[5] Z. Zhang, E. -H. Yang and V. K. Wei, " The redundancy of source coding with a fidelity criterion," to appear in *IEEE Trans. Inform. Theory.*

The Association Scheme of Quaternary Preparata Codes

Alexis Bonnecaze
bonnecaz@essi.fr
Iwan M. Duursma
duursma@pascal.uprr.pr

We use the association scheme of the quaternary Preparata codes to construct other quaternary codes and to realize difference sets on the additive group of the Galois ring of characteristic four.

The Association scheme

Let R be the Galois ring of degree m over $\mathbf{Z}_4$. It is of size 4^m and it contains the subset T of all squares in R, that is of size 2^m. Following [1], the quaternary Preparata code of order m is the set of all $\mathbf{Z}_4$ valued vectors $(c_t : t \in T)$ of length 2^m, such that

$$\sum_{t \in T} c_t = \sum_{t \in T} c_t t = 0.$$

The syndromes of the Preparata code form the additive group $\mathbf{Z}_4 \times R$. We define a partition $\mathbf{\Omega} = \{\Omega_0 = 0, \Omega_1, \ldots, \Omega_r\}$ of the elements of $\mathbf{Z}_4 \times R$, such that two syndromes are in the same class if and only if they have the same coset weight distribution. The message vectors for the dual code form the same group $\mathbf{Z}_4 \times R$. We define a partition $\mathcal{E} = \{\mathcal{E}_0 = 0, \mathcal{E}_1, \ldots, \mathcal{E}_s\}$ of the elements of $\mathbf{Z}_4 \times R$, such that two message vectors are in the same class if and only if the corresponding codewords have the same complete weight. Theorem 6 in [2] shows that $\mathbf{\Omega}$ and $\mathcal{E}$ satisfy the axioms of an association scheme and its dual, with $r = s = 9$. We give a partial description,

$$\left\{\begin{array}{lcl} 0 \times R &=& \Omega_0 \cup \Omega_3 \cup \Omega_9, \\ 1 \times R &=& \Omega_1 \cup \Omega_7, \\ 2 \times R &=& \Omega_4 \cup \Omega_5 \cup \Omega_6, \\ 3 \times R &=& \Omega_2 \cup \Omega_8. \end{array}\right. \qquad \left\{\begin{array}{lcl} 0 \times 2R &=& \Omega_0 \cup \Omega_9, \\ 1 \times T &=& \Omega_1, \\ 2 \times 2R &=& \Omega_4, \\ 3 \times -T &=& \Omega_2. \end{array}\right.$$

$$\left\{\begin{array}{lcl} \mathbf{Z}_4 \times 0 &=& \mathcal{E}_0 \cup \mathcal{E}_3 \cup \mathcal{E}_4 \cup \mathcal{E}_9, \\ \mathbf{Z}_4 \times 2R \backslash 0 &=& \mathcal{E}_5 \cup \mathcal{E}_6, \\ \mathbf{Z}_4 \times R \backslash 2R &=& \mathcal{E}_1 \cup \mathcal{E}_2 \cup \mathcal{E}_7 \cup \mathcal{E}_8. \end{array}\right.$$

Codes

Let $C(S)$ denote the code whose generator matrix has as columns the elements of the set S. In particular, the code $C(\Omega_1)$ stands for the Kerdock code and its dual $C(\Omega_1)^\perp$ for the Preparata code. For the set S, we may take other classes in the partition $\mathbf{\Omega}$, or combinations of them. By doing so, the weight distribution of $C(S)$ is immediate from the eigenvalues of the association scheme. Many of these codes are good, but among the binary images we found no improvements on existing codes. Let m be odd. The binary images of the codes $C(\Omega_8)$ and $C(\mathcal{E}_8)$ have the same parameters. The length $n = 2(4^m - 2^m)$, the size $M = 4^{m+1}$, and the minimum distance $d = 4^m - 2^m - 2^{(m-1)/2}$. Their weight distribution is different. A linear code with parameters $[112, 8, 54]$ (the case $m = 3$) is given in [3].

weight	frequency
0	1
$2^{2m} - 2^m - 2^{(m-1)/2}$	$2^{2m+1} - 2^{m+1}$
$2^{2m} - 2^m$	$2^{m+2} - 2$
$2^{2m} - 2^m + 2^{(m-1)/2}$	$2^{2m+1} - 2^{m+1}$
$2^{2m+1} - 2^{m+1}$	1

Table 1: The binary code $C(\Omega_8)$.

weight	frequency
0	1
$2^{2m} - 2^m - 2^{(m-1)/2}$	$2^{2m+1} - 2^{m+1}$
$2^{2m} - 2^m$	$2^{2m+1} - 2^m$
2^{2m}	$2^m - 1$
$2^{2m} - 2^m + 2^{(m-1)/2} \cdot (2^m - 1)$	2^{m+1}

Table 2: The binary code $C(\mathcal{E}_8)$.

Difference Sets

The partition $\mathbf{\Omega}$ reveals nice combinatorial properties of the Galois ring in characteristic four. The subset of squares in a Galois ring over $\mathbf{Z}_4$ defines a relative difference set with parameters $(2^m, 2^m, 2^m, 1)$ [2]. A relative difference set with these parameters was constructed in [4] without reference to Galois rings. The group law used in [4] differs only slightly from the addition law in a Galois ring. Furthermore, the subset of sums of two different squares in a Galois ring over $\mathbf{Z}_4$ defines a perfect difference set with classical parameters $(2^{2m}, 2^{2m-1} - 2^{m-1}, 2^{2m-2} - 2^{m-1})$. Difference sets with these parameters are known to exist on every group that is a product of groups of order four [5]. However, known constructions of such sets appear to be formulated without reference to Galois rings over $\mathbf{Z}_4$. The claims are conveniently expressed by relations on characteristic functions in the group ring of $\mathbf{Z}_4 \times R$. The relation $[\Omega_1][\Omega_1] = [\Omega_4] + 2[\Omega_5]$ defines Ω_5 and completes the description of $\mathbf{\Omega}$. We have

$$[\Omega_1][-\Omega_1] = 2^m\,[\Omega_0] + [\Omega_3].$$

$$[\Omega_5][-\Omega_5] = (2^{2m-1} - 2^{m-1})\,[\Omega_0] + (2^{2m-2} - 2^{m-1})\,([\Omega_3] + [\Omega_9]).$$

References

[1] R. Hammons, P.V.Kumar, A.R. Calderbank, N.J.A Sloane, and P. Solé. Kerdock, Preparata, Goethals and Other Codes are linear over Z4. *IEEE Transactions on information theory*, 40:301–319, 1994.

[2] A. Bonnecaze, and I.M. Duursma. Translates of Linear Codes over $\mathbf{Z}_4$. Submitted for publication, 1995.

[3] H.C.A. van Tilborg. On quasi-cyclic codes with rate 1/m. *IEEE Transactions on information theory*, 24:628-630, 1978.

[4] D. Jungnickel. On automorphism groups of divisible designs. *Can. J. Math.*, 34:257–297, 1982.

[5] M. Hall, Jr. Difference Sets. Combinatorics (M. Hall, Jr. and J.H. van Lint eds.), 321-346, D. Reidel Publ., 1975.

[6] R.A. Liebler and R.A. Mena. Certain distance-regular digraphs and related rings of characteristic 4. *J. Comb. Theory*, 47:111–123, 1988.

[7] M. Yamada. Distance-Regular Digraphs of Girth 4 Over an Extention Ring of Z/4Z. *Graphs and Combinatorics*, 6:381–394, 1990.

A Class of Frequency Hop Codes with Nearly Ideal Characteristics for Multiple-target Recognition

Oscar Moreno[1]
Department of Mathematics and Computer Science,
University of Puerto Rico,
Rio Piedras, Puerto Rico, 00931.
and
Svet Marić
Department of Electrical Engineering
University of Cambridge
Cambridge CB2 1PZ, U.K.

Abstract: The problem we address is that of obtaining the velocity and the distance from us of one or several targets. The case of only one target has been developed by S. W. Golomb and H. Taylor in 1982 and 1984 and also by O. Moreno, R. A. Games and H. Taylor in 1993. In this paper, we deal with the case of multiple-target recognition and develop a class of frequency hop codes with nearly ideal characteristics.

1 Introduction

The problem of construction of code sequence with good auto- and cross-correlation properties for multiple access communications has always received a lot of attention among the researchers in various communication fields. M-sequences and sequences derived from them solve the problem in the one dimensional space. However, the problem of sequence construction becomes much more complicated when extended to two dimensions as is the case in 1) multiuser radar and sonar systems (the horizontal shift is time and vertical Doppler), and 2) FH-CDMA mobile radio (the horizontal shift is again time and the vertical shift is this time the user's message).

For this 2-D case there are various algebraically constructed FH code families, the most famous one being the Costas arrays [2] who possess "ideal" 2-D auto-correlation properties (ideal for use in a single user environment). Here "ideal" refers to having at most one coincidence between the original 2-D matrix representing the FH code, and its shifted replica, for any possible time-Doppler shift. Unfortunately, Costas arrays [4] do not have good 2-D cross-correlation properties and hence are not always suitable for multiple access communications.

On the other hand the linear congruence FH codes constructed as an extension of the chirp signal have ideal 2-D cross-correlation but poor 2-D auto-correlation functions - so for instance, in the radar case for the unknown range of the target it is almost impossible to estimate its corresponding velocity.

Other known families of FH codes trade the ideal 2-D auto- or cross-correlation properties for a better overall performance. For instance the so called hyperbolic congruence

---

[1] Work partially supported by NSF grants CDA-9417362, component IV of the EPSCoR of Puerto Rico Grant and the ARO grant for Cornell MSI.

FH codes [5] have at most two coincidences both in their 2-D auto- and cross correlation functions.

One major disadvantage of all the above FH code families is the fact that their construction (except for a limited number of Costas arrays) is limited to finite fields over a prime, $GF(p)$.

In this paper we introduce new families of FH codes with good 2-D correlation properties whose construction is not limited to $GF(p)$ but remains valid for $GF(p^n)$. The codes are in fact a cross between the so called Lempel-Golomb constructions for Costas arrays and various families of n-th order congruence codes.

Our construction produces families of FH codes that falls in what are called extended sonar and Costas sequence (see [8]). In the case of ordinary sonar and Costas sequence exactly one signal at every time is sent, on the other hand in the case of extended sonar sequences *at most* one signal is sent at every time, or in other words, one can choose not to send any signal in some time slots. The problem for this application is given n frequencies, construct and n by m sequence for m as large as possible. As shown in [8] with the extended sonar sequence one can obtain larger sequences with the same range of frequencies than with the ordinary sonar sequence. This last point also means that one can use the same equipment for both sequences making the extended sonar sequence cost effective.

The paper is organized as follows. In the next section, for completeness, we first give some necessary definitions and also the constructions of Welch (WC) and Lempel-Golomb (LC) arrays, and quadratic and hyperbolic congruence codes. In the third section we give the constructions of our new FH code families and also describe their 2-D auto- and cross-correlation properties.

2 Preliminaries

Frequency Hop codes in general are divided in two groups based on the implemented signal processing at the receiver. In his original work on FH radar and sonar systems J. Costas [1] pointed out the importance of the FH codes having one and only one pulse per each time slot and frequency channel. We call those type of FH codes full FH codes. The most famous example of full FH codes are the Costas arrays.

Definition 1: [2] A Costas array is an $N \times N$ permutation matrix with the property that vectors connecting two 1's are all distinct vectors. (That is no two vectors are equal in both amplitude and slope.)

An extensive study of the algebraic constructions of Costas arrays is given in [5] and [2]. All known constructions producing a large number of Costas arrays of equal length are based on the properties of finite fields and primitive roots.

Theorem 1 (Welch):[2] Let α be a primitive root of an odd prime p. Then the array with

$$a_{k,y(k)} = 1,\ 1 \leq k, y(k) \leq p-1 \tag{1}$$

if and only if

$$y(k) = \alpha^k \, (mod p),\ 1 \leq k \leq p-1, \tag{2}$$

otherwise $a_{k,y(k)} = 0$ is a Costas array.

---

[2] The proofs of the theorems in this sections can be found in [5] or [2].

Note that the extensions of this construction have been found by Golomb and Taylor [3], see also Moreno [6][3].

Theorem 2 (Lempel-Golomb): Let α and β be primitive roots of the finite field, $GF(q)$, $q = p^n$, p a prime. Then the $(q-2) \times (q-2)$ permutation matrix with $a_{k,y(k)} = 1$ if and only if

$$y(k) = log_\beta(1 - \alpha^k),\ 1 \leq k \leq (q-2) \tag{3}$$

is a Costas array.

It is important to note the difference between the finite fields in the two theorems. Theorem 1 is valid only over $GF(p)$ whereas theorem 2 is valid also for $GF(p^n)$ which then requires the representation of the elements through polynomials.

By definition the number of hits in the 2-D auto-correlation function of Costas arrays is ideal. However the multi-user environment and rather poor 2-D cross-correlation properties of Costas arrays necessitated the construction of other FH families like the n-th order congruence codes.

Definition 2: Let $J(p)$ denote the finite field over a prime p, and let $\tilde{J}(p) = J(p) - \{0\}$ Then the n-th order congruence FH code of length p is defined by:

$$y(k) = ak^n\ (mod p)\ 0 \leq k \leq p\ and\, some\, a \in \tilde{J}(p) \tag{4}$$

Theorem 3: An n-th order congruence code constructed over the field of prime p, is a full FH code when $d = (\phi(p), n) = 1$ where ϕ denotes the Euler ϕ function and $d = (x, y)$ is the greatest common divisor of x and y.

Theorem 4: A n-th order congruence FH code $(n > 1)$ of length p has at most $n-1$ hits in its 2-D auto-correlation function (except for the (0,0) shift and at most n hits in its 2-D cross-correlation function formed with any other code from the same family.

A special case of (4) is the case $n = p - 2$. Then the constructed codes have at most two hits in both the 2-D auto- and 2-D cross-correlations functions, and represent the optimal full FH codes in terms of simultaneous number of hits in the 2-D auto- and cross-correlation functions. Another interesting case for our purposes is the case of $n = 2$. The obtained quadratic congruence FH codes are not full (see Theorem 3) but have at most 1 hit in the 2-D auto-correlation and at most 2 hits in the 2-D cross-correlation function.

3 New class of frequency hop codes

To define our new class of codes, consider as before α, β primitive roots of the finite field GF(q) and $P_n(x)$ the set of all polynomials of degree $\leq n$ over GF(q). If $f(x) \in P_n(x)$ we will define an equivalence relation so that $f(x)$ is equivalent to $bf(ax)$ where b, a are arbitrary non-zero elements of GF(q). Consider now $E_n(x)$ a subset of $P_n(x)$ containing a set of representatives of the equivalence relation, excluding all the polynomials which are non-trivially self-equivalent. In other words any element of $E_n(x)$ is not equivalent to any other element of $E_n(x)$, and is also not self-equivalent. Furthermore $E_n(x)$ is as large as possible given the above mentioned properties under the equivalence relation.

We define now

$$y_f(k) = \log_\beta f(\alpha^k) \qquad 1 \leq k \leq q-1$$

---

[3] In general, every array obtained by a multiplication of the original array (given by (2)), by an integer is a Welch array as well. But those arrays represent just shifted versions of the original array.

where $f \in E_d(x)$ with α, β primitive elements over $\mathrm{GF}(q)$. Note that the array defined using y_f must have empty columns whenever $f(\alpha^i) = 0$, and therefore falls in the category of extended patterns define in [8]. With the above comments we have our main result:

Theorem 5: The codes defined by y_f for $f \in E_d(x)$ have at most d hits in its 2-D auto-correlation and 2-D cross-correlation function.

Remark: Note that $E_d(x)$ has approximately q^{d-1} elements.

References

[1] J. P. Costas. A Study of a Class of Detection Waveforms Having Nearly Ideal Range-Doppler Ambiguity Properties. In *Proceedings of the IEEE, vol 72, no 8* pp 996-1009, August 1984.

[2] S.W. Golomb and H. Taylor. Construction and Properties of Costas arrays. In *Proceedings of the IEEE, vol. 72, no. 9* September 1984.

[3] S.W. Golomb and H. Taylor. Two dimensional synchronization patterns for minimum ambiguity. In *"IEEE Trans. Information Theory, vol. IT-28 no. 4* July, 1982.

[4] S. V. Maric and I. Seskar and E. L. Titlebaum. On cross-ambiguity Properties of Welch-Costas arrays. In *"IEEE Transactions on Aerospace and Electronic Systems,* October 1994.

[5] S.V. Maric and E.L. Titlebaum. A Class of Frequency Hop Codes with Nearly Ideal Characteristics for Use in Multiple Access Spread Spectrum Communications and Radar and Sonar Systems In *IEEE Transactions on Communications,* September 1992..

[6] O. Moreno. A shifting property of some Costas sequences. In *Ars Combinatoria* June, 1994.

[7] O. Moreno, R. A. Games and H. Taylor, *Sonar Sequences from Costas arrays and the best known sonar sequences with up to 100 symbols,* IEEE Transactions on Information Theory, 39:6 (1993), pp. 1985–1989.

[8] O. Moreno, S. W. Golomb and C. J. Corrada. Extended Sonar and Costas sequences. Pre-print.

Capacity and Dynamic Resource Allocation in Broadcast Fading Channels

Andrea Goldsmith*
California Institute of Technology
Mail Code 116-81, Pasadena, CA 91125

Abstract

This paper explores the capacity bounds and the dynamic allocation of resources in multiuser fading channels. We first obtain the single-user capacity of an AWGN fading channel, both with and without power adaptation. We show that channel side information at the transmitter does not increase capacity unless the transmit power is also adapted, although it does significantly reduce the decoder complexity. Moreover, optimal power adaptation does not yield a significant capacity increase relative to constant power transmission in either Rayleigh or log-normal fading. However, inverting the channel results in a very large capacity penalty. The tradeoff in performance of these three power adaptation techniques is one of complexity versus capacity. We then extend these results to obtain the multiuser capacity region for broadcast fading channels with dynamic power and bandwidth adaptation. This yields an analytical framework to compare the efficiency of channel sharing using time-division, frequency-division, and code-division. We conclude that code-division with interference cancellation is optimal, but both time- and frequency-division have superior performance if interference cancellation is not exploited. Moreover, using spread spectrum to implement code-division results in some performance loss relative to narrowband superposition coding. Finally, if the received power of all users is kept constant, then code-division with interference cancellation does no better than the other techniques.

1 Introduction

Wireless communication systems require efficient use of the limited available spectrum and the underlying fading channel. Methods to divide the spectrum among many users include frequency-division, time-division, code-division, and hybrid combinations of these methods. Although there have been many performance comparisons of these techniques in the literature [1, 2], these studies generally required some assumptions about the system specifications. We remove system considerations by evaluating the multiuser Shannon capacity of a fading channel for these different spectrum-sharing techniques.

We first describe results for the capacity of a single-user fading channel, when the channel fade level is tracked by both the transmitter and receiver, and by the receiver alone. We do not consider the case when the channel fading is not known to either the transmitter or receiver: the capacity of this channel assuming a Markov channel variation was derived in [3]. For memoryless channels without power adaptation, side information at the transmitter does not increase capacity, although it greatly reduces the decoder complexity. When the channel fade value is known at

---

*This work was supported by the Office of Naval Research under Grant No. NAV-5X-N149510861 and by NSF CAREER Development Award No. NCR-9501452.

the transmitter, then capacity can be increased by adapting three parameters to the channel variation: transmit power, data rate, and coding scheme. We compare the nonadaptive and adaptive transmission methods in both log-normal and Rayleigh fading, and obtain the somewhat surprising result that the capacity does not significantly increase with adaptation to the channel fading. However, the capacity of the nonadaptive method requires a maximum-likelihood decoder for all possible channel realizations, which is prohibitively complex to implement in practice. Although the adaptive method must transmit at a variable rate and power, its decoding technique does not depend on the channel realization. Thus, the tradeoff for single-user channels between adaptive and nonadaptive techniques is one of complexity: assuming that the channel is estimated at the receiver, the adaptive technique requires a feedback path between transmitter and receiver and some complexity in the transmitter, while the nonadaptive technique places a much higher complexity burden on the receiver. We examine these complexity issues in more detail in [4] for fixed and variable-rate coded MQAM.

We next extend our single-user capacity analysis to broadcast channels. The rate region for a two-user broadcast channel with independent fading at each receiver is evaluated. We will see that code-division with interference cancellation and adaptive power allocation achieves the maximum rate region. Moreover, if the received power of all users is the same, then there is no capacity difference for the different spectrum sharing techniques. In addition, spread spectrum implementation of code-division yields a capacity penalty which is a function of the spreading gain. Finally, without interference cancellation, code-division is generally inferior to both frequency-division and time-division.

2 Single-User Fading Channels

Consider a discrete-time channel with stationary and ergodic time-varying gain $\sqrt{L\gamma[i]}$ and additive white Gaussian noise $n[i]$. Let S denote the average transmit signal power, $N_0/2$ denote the noise density of $n[i]$, B denote the received signal bandwidth, and L denote the propagation loss in the absence of fading. With appropriate scaling of S we can set $L = 1$. The average received SNR is then S/N_0B, which we assume is known to both the transmitter and receiver.

The system model is illustrated in Figure 1. We assume that the channel power gain $\gamma[i]$ is available to the receiver at time i. We also consider the case when $\gamma[i]$ is available to both the receiver and transmitter at time i. This allows the transmitter to adapt to the channel gain at time i, and is a reasonable model for a slowly-varying channel with channel estimation and transmitter feedback.

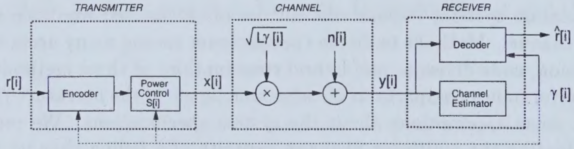


Figure 1: System Model

2.1 Side Information at the Transmitter and Receiver

The capacity of time-varying channels with side information at both the transmitter and receiver and at the receiver alone was originally consider by Wolfowitz [5]. He showed that in the former case, the capacity is just the weighted average of the channel capacity at each point in time. We now apply this capacity formula to an AWGN fading channel[1] Specifically, assume an AWGN fading channel with time-varying stationary and ergodic channel gain $\gamma[i]$. Let $C(\gamma)$ denote the capacity of an AWGN channel with SNR γ, and $p(\gamma) = p(\gamma[i] = \gamma)$ denote the probability distribution of the time-varying SNR. The capacity of this channel with transmitter and receiver side information is given by

$$C = \int_{\gamma} C(\gamma)p(\gamma)d\gamma. \tag{1}$$

Suppose we also allow the transmit power $S(\gamma)$ to vary, subject to an average power constraint S:

$$\int S(\gamma)p(\gamma)d\gamma \; = \; S. \tag{2}$$

Let

$$C_{\gamma}(S(\gamma), B) \triangleq B \log\left(1 + \frac{S(\gamma)\gamma}{N_0 B}\right) \tag{3}$$

denote the capacity of an AWGN channel with bandwidth B, instantaneous SNR γ and average received power $S(\gamma)$. From (1), the capacity of the fading channel with power adaptation is then

$$C(S, B) = \max_{S(\gamma): \int S(\gamma)p(\gamma)d\gamma = S} \int B \log\left(1 + \frac{S(\gamma)\gamma}{N_0 B}\right) p(\gamma)d\gamma. \tag{4}$$

Using Lagrange multipliers, it is easily shown that the power adaptation which maximizes (4) is

$$\frac{S(\gamma)}{S} = \begin{cases} \frac{1}{\gamma_0} - \frac{1}{\gamma} & \gamma \geq \gamma_0 \\ 0 & \gamma < \gamma_0 \end{cases} \tag{5}$$

for some "cutoff" value γ_0. If the received signal power is below this level, then no power is allocated to data transmission. Since γ is time-varying, the maximizing transmit power distribution (5) is a "water-pouring" formula in time [9] that depends on the fading statistics only through the cutoff value γ_0.

Substituting (5) into (2), we can determine γ_0 by numerically solving

$$\int_{\gamma_0}^{\infty} \left(\frac{1}{\gamma_0} - \frac{1}{\gamma}\right) p(\gamma)d\gamma = 1. \tag{6}$$

Once γ_0 is known, we substitute (5) into (4) to get

$$C(S, B) = B \int_{\gamma_0}^{\infty} \log\left(\frac{\gamma}{\gamma_0}\right) p(\gamma)d\gamma. \tag{7}$$

The channel coding and decoding strategy which achieves this capacity can be modeled as a "time diversity" system with multiplexed input and demultiplexed output. There is a pair of input and output ports associated with each value of the instantaneous power gain $\gamma[i]$. When $\gamma[i] = \gamma_j$, the corresponding pair of ports are connected through the time-invariant channel with gain γ_j. Thus, we may regard the single time-varying channel as a series of time-invariant channels in parallel, where the jth channel only operates when $\gamma[i] = \gamma_j$. Suppose an encoder and codewords x_j are optimally designed for the time-invariant channel associated with γ_j. By Shannon's theorem [6], we can develop these x_j codewords with average power $S(\gamma_j)$ that achieve channel capacity for the AWGN time-invariant channel with gain γ_j. The codewords associated with each γ_j can then be time-multiplexed to form a single codeword x, which has variable data rate and power. The average power of the multiplexed codebook is then $\int p(\gamma)S(\gamma)d\gamma$. In [8] we show that these codewords achieve the capacity (4), and that no other set of codewords can achieve a rate bigger than this capacity with an arbitrarily small probability of error.

---

[1] Wolfowitz's derivation assumed that the channel varied over a finite set of discrete memoryless channels, however his result can be extended to continuous sets [8].

2.2 Side Information at the Receiver

In [7] it was shown that if the channel variation satisfies a compatibility constraint, then the capacity of the channel with side information at the receiver only is also given by the average capacity formula (1). The compatibility constraint is satisfied if the input distribution which maximizes mutual information is the same regardless of the channel state. In this case, the side information at the transmitter does not increase capacity, although it greatly reduces the complexity of the decoder.

The fading channel can be modeled as an AWGN channel with varying noise statistics. If the transmit power is fixed at S then, for any noise density, the input distribution which achieves capacity is an i.i.d. Gaussian distribution with average power S. Thus, without power adaptation, the fading AWGN channel satisfies the compatibility constraint of [7]. The channel capacity with receiver side information only is thus given by

$$C(S,B) = \int B \log\left(1 + \frac{S\gamma}{N_0 B}\right) p(\gamma) d\gamma. \tag{8}$$

Equation (8) implies that without power adaptation, information about the channel fade level cannot be used at the transmitter to increase capacity. This is because the transmitter uses the same optimal input distribution regardless of the channel state, so it need not adapt its strategy to the channel variation.

2.3 Channel Inversion

In this section we assume transmitter side information, as in §2.1, with a different power control strategy. The goal is to maintain a constant received power, so the transmitter exactly compensates for the signal fading. Thus, the power control policy is $S(\gamma)/S = P_R/\gamma$, where P_R equals the received signal-to-noise ratio. The channel then appears as a time-invariant AWGN channel with SNR $= P_R$. The constant P_R is determined by the transmit power constraint (2):

$$\int \frac{P_R}{\gamma} p(\gamma) = 1 \qquad \Longrightarrow \qquad P_R = \frac{1}{\mathrm{E}[1/\gamma]}. \tag{9}$$

The channel capacity with this power control strategy is derived from the capacity of an AWGN channel with receive power P_R:

$$C(S,B) = B \log\left[1 + P_R\right] = \log\left[1 + \frac{1}{\mathrm{E}[1/\gamma]}\right]. \tag{10}$$

This form of power control is common in cellular systems with near-far interference imbalances [1]. It is also the simplest scheme to implement, since the encoder and decoder are designed for AWGN, independent of the fading statistics.

We also consider a modified inversion policy that only compensates for fading above a certain cutoff fade depth γ_0:

$$\frac{S(\gamma)}{S} = \begin{cases} \frac{P_R}{\gamma} & \gamma \geq \gamma_0 \\ 0 & \gamma < \gamma_0 \end{cases}. \tag{11}$$

Since the channel is only used when $\gamma \geq \gamma_0$, the power constraint (2) yields $P_R = 1/\mathrm{E}_{\gamma_0}[1/\gamma]$, where

$$\mathrm{E}_{\gamma_0}[1/\gamma] \triangleq \int_{\gamma_0}^{\infty} \frac{1}{\gamma} p(\gamma) d\gamma. \tag{12}$$

The capacity is thus

$$C(S,B) = B \log_2\left[1 + \frac{1}{\mathrm{E}_{\gamma_0}[1/\gamma]}\right] P(\gamma \geq \gamma_0), \tag{13}$$

where $P(\gamma \geq \gamma_0) = \int_{\gamma_0}^{\infty} p(\gamma) d\gamma$. To get the maximum efficiency for this truncated channel inversion policy, we must maximize (13) relative to γ_0. Alternatively, we can set γ_0 to achieve a desired outage probability $P_{out} = p(\gamma < \gamma_0)$. We will see in the next section that without truncation, the channel inversion policy yields a capacity of zero for Rayleigh fading channels.

2.4 Numerical Results

Figure 2 shows a plot of (4), (8),(10), and (11) (for different γ_0 criteria) as a function of SNR for log-normal fading with $\sigma = 8$dB. For Rayleigh fading, E$[1/\gamma]$ is infinite: thus, the spectral efficiency with the channel inversion policy is zero. Figure 3 shows the capacity of the other policies in Rayleigh fading. There are several observations worth noting in these figures. First, in both types of fading, side information and power adaptation at the transmitter yields a negligible increase in capacity relative to the nonadaptive technique. This result is somewhat surprising: it means that the only benefit of adaptive transmission is a complexity reduction in the decoding at the receiver. In addition, truncated channel inversion exhibits a one to three dB rate penalty, and channel inversion without truncation ($P_{out} = 0$) yields a very large capacity loss (infinite in the case of Rayleigh fading). We also see that maintaining a fixed outage probability at low SNRs causes some loss relative to the maximum C/B obtained with truncated inversion, but this loss diminishes with increasing SNR.

The results of these plots can be interpreted as the tradeoff between capacity and complexity. The adaptive policy with transmitter side information requires more complexity in the transmitter (and a feedback path between the receiver and transmitter to obtain the side information). However, the decoder in the receiver is relatively simple. The nonadaptive policy does not require a feedback path, however its decoder must perform maximum-likelihood sequence detection for all possible channel realizations. The channel inversion policy uses a code designed for an AWGN channel, and is therefore the least complex technique (although it does require a feedback path for the transmitter to obtain the channel side information). However, the inversion policy has potentially huge capacity losses relative to the other techniques. As a caveat, it is not clear if the complexity associated with codes that achieve Shannon capacity will be proportional to the complexity of practical coding techniques for the adaptive and nonadaptive techniques. This complexity tradeoff for trellis-coded MQAM is currently being investigated in [4].

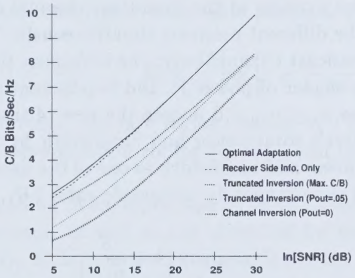


Figure 2: Capacity in Log-Normal Fading

3 Rate Regions for Broadcast Channels

When several users share the same channel, the channel capacity can no longer be characterized by a single number. At the extreme, if all but one user occupies the channel, then the single-user capacity results of the previous section apply. However, since there is an infinite number of ways to "divide" the channel between many users, the multiuser channel capacity is characterized by a *rate region*, where each point in the region is a vector of achievable rates that can be maintained by all the users simultaneously. The set of all achievable rates is called the *capacity region* of the multiuser system.

In this section we analyze the capacity of the broadcast channel in fading and AWGN. We begin by first reviewing results from [6] for the time-invariant AWGN broadcast channel using

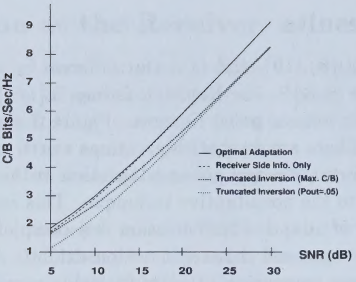


Figure 3: Capacity in Rayleigh Fading

code-division with successive decoding (interference cancellation), time-division, and frequency-division. We also derive the rate region of code-division without interference cancellation, and the capacity penalty as a function of spreading gain when the code-division is implemented using spread spectrum coding. We then combine the single-user fading capacity analysis of §2 with the time-invariant rate region analysis to obtain the broadcast channel capacity in fading.

The maximum rate region in both AWGN and fading is achieved using superposition code-division with interference cancellation. We will elaborate on this technique in the following sections. We will also show that the rate region of code-division without interference cancellation is inferior to all the other spectrum-sharing techniques.

3.1 The AWGN Broadcast Channel

The broadcast channel consists of one transmitter sending information to many receivers over a common channel. The capacity region of the broadcast channel characterizes how much information can be conveyed to the different receivers simultaneously. We consider rate regions for a two-user discrete AWGN broadcast channel only; the extension to multiple users is straightforward [11]. Thus, there is one sender of power P, and two distant receivers, each with AWGN of power n_i, $i = 1, 2$. We assume $n_1 \leq n_2$, and denote the rate of transmission to receiver i by R_i.

We denote the transmitter's total power and bandwidth by S and B, respectively. If the transmitter allocates all the power and bandwidth to one of the users, then the other user receives no data; therefore, the set of simultaneously achievable rates (R_1, R_2) includes the pairs $(C_1, 0)$ and $(0, C_2)$, where

$$C_i = B \log\left[1 + \frac{S}{n_i B}\right]. \tag{14}$$

These two points bound the multiuser capacity region. We now consider rate pairs in the interior of the region, which are achieved using more equitable methods of dividing the channel resources.

In time-division, the full power and bandwidth is allocated to user 1 for a fraction τ of the total transmission time, and then to user 2 for the remainder of the transmission. This time division scheme achieves the straight line in Figure 4 corresponding to the rate pairs $(\tau C_1, (1 - \tau)C_2), 0 \leq \tau \leq 1$. An alternative approach for spectrum-sharing is frequency division. In this method the ith user is allocated power S_i and bandwidth B_i of the total, so $S_1 + S_2 = S$ and $B_1 + B_2 = B$. The set of achievable rates (R_1, R_2), for fixed S_i and B_i, is then given by

$$R_i = B_i \log\left[1 + \frac{S_i}{n_i B_i}\right]. \tag{15}$$

It was shown by Bergmans [11] that, for n_1 strictly less than n_2 and any fixed frequency division (B_1, B_2), there exists a range of power allocations (S_1, S_2) whose corresponding rate pairs dominate a segment of the time division rate region. Thus, the rate regions achievable through time division can always be exceeded by optimizing both the frequency and power allocation in (15). This is shown in Figure 4 for channels wth SNRs which differ by 3dB and by 20dB. It can be shown [11] that when the two channels have the same SNR, time-division and frequency-division have the same rate regions. The intersection point of the time- and frequency-division curves in Figure 4 corresponds to a power allocation S_i that is proportional to the bandwidth B_i. Moreover, it was also shown in [11] that time-division with different power allocated to the two users can achieve the same rate region as frequency-division.

The rate region using superposition coding and successive decoding was derived in [11] to be the convex hull of rate pairs

$$R_1 = B \log \left[1 + \frac{\alpha_1 S}{n_1 B}\right], \quad R_2 = B \log \left[1 + \frac{\alpha_2 S}{n_2 B + \alpha_1 S}\right], \tag{16}$$

where $\alpha_1 + \alpha_2 = 1$. The intuitive explanation for (16) is the following. Since $n_1 < n_2$, user 1 correctly receives all the data transmitted to user 2. Therefore, user 1 can correctly decode and then subtract out user 2's message, then decode its own message. User 2 cannot decode the message intended for user 1, since it has a less-favorable channel; thus, user 1's message, with power $\alpha_1 S$, contributes an additional noise term to user 2's received message.

The rate region defined by (16) was shown in [12] to strictly dominate the regions achievable through either time or frequency division, when $n_1 < n_2$. Moreover, Bergmans shows in [12] that (16) defines the capacity region, i.e., the maximum achievable set of rate pairs. However, superposition coding is superior only when $n_1 \neq n_2$; otherwise, all the spectrum-sharing methods we have described have the same rate region [12].

Code-division in cellular systems is commonly implemented using spread spectrum [1]. With spread spectrum codes, the signal bandwidth is spread by a factor $G > 1$ typically called the spreading gain. The cross correlation between the respective codes is then proportional to $1/G$, so that interference between codes is reduced by a factor of G. Since the signal bandwidth is also increased by this factor, the rate region achievable through spread-spectrum code-division is given by

$$R_1 = \frac{B}{G} \log \left[1 + \frac{\alpha_1 S}{n_1 B}\right], \quad R_2 = \frac{B}{G} \log \left[1 + \frac{\alpha_2 S}{n_2 B + \alpha_1 S/G}\right], \tag{17}$$

By the convexity of the log function, the rate region defined by (17) for $G > 1$ is smaller than the rate region obtained using superposition coding ($G = 1$), and the degradation increases with increasing values of G.

In practice, successive decoding of superposition or spread-spectrum codes adds complexity and delay in the decoding process, as well as the potential for feedback errors when user 2's message is not decoded properly. Most commercial spread spectrum receivers don't use successive decoding: they treat all messages intended for other users as noise, resulting in a rate region (R_1, R_2) satisfying

$$R_1 = \frac{B}{G} \log \left[1 + \frac{\alpha_1 S}{n_1 B + \alpha_2 S/G}\right], \quad R_2 \leq \frac{B}{G} \log \left[1 + \frac{\alpha_2 S}{n_2 B + \alpha_1 S/G}\right], \tag{18}$$

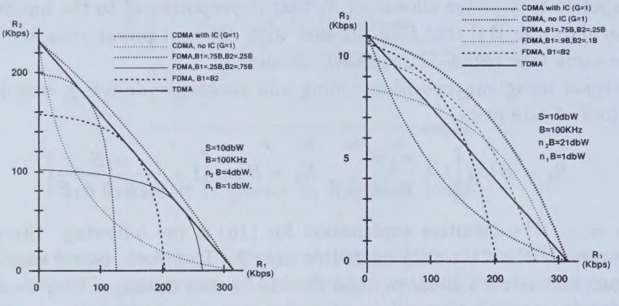
where $\alpha_1 + \alpha_2 = 1$. From the convexity of the log function, $G = 1$ maximizes this rate region, and the rate region decreases as G increases.

Taking the second partial derivatives in (18), we get that for any $G \geq 1$,

$$\frac{\partial^2 R_2}{\partial^2 R_1} = \frac{\partial R_1}{\partial \alpha_1} \frac{\partial^2 R_2}{\partial^2 \alpha_1} - \frac{\partial R_2}{\partial \alpha_1} \frac{\partial^2 R_1}{\partial^2 \alpha_1} \geq 0. \tag{19}$$

Thus, the rate region (R_1, R_2) is convex with end points C_1 and C_2, as shown in Figure 4. Therefore, *both time-division and frequency-division always dominate superposition coding without successive decoding.*

The rate regions for time-division, frequency-division, and code-division (superposition coding) with and without interference cancellation are illustrated in Figure 4. This figure shows superposition coding with $G = 1$: spread spectrum codes with $G > 1$ will have smaller rate regions.



4a: 3dB difference in noise densities **4b: 20dB difference in noise densities**

Figure 4: Broadcast Channel Rate Region.

3.2 The Broadcast Channel with Fading and AWGN

We now combine the results from §2 with the rate region analysis above to obtain the rate region for the broadcast channel when both users experience signal fading. We will focus in this section on the case where side information is available at both the transmitter and receiver, i.e. the case where the transmitter can adaptively allocate both power and bandwidth to the two users as the channel conditions change.

The two-user broadcast channel with fading has one transmitter with time-varying power $S[i]$ and bandwidth B, and two receivers with time-varying noise density $n_j[i], j = 1, 2$. Side information at the receiver implies that the jth receiver tracks the noise level $n_j[i]$. Side information at the transmitter implies that the transmitter also tracks both $n_1[i]$ and $n_2[i]$. In this case the transmitter can vary its instantaneous power $S[i]$ relative to $n_1[i]$ and $n_2[i]$, subject only to the average power constraint S. For frequency-division, it can also vary the bandwidth $B_j[i]$ allocated to each user, subject to the constraint $B_1[i] + B_2[i] = B \ \forall i$.

With time-division, the average transmit power S and total bandwidth B is allocated to the first user over the time interval $[0, \tau T]$ and to the second user over the time interval $[\tau T, T]$. This method reduces the two-user channel to two orthogonal time-varying single-user channels, so we can use the time-varying channel capacity results in §2. Thus, we can achieve any rate point

$$(R_1, R_2) = (\tau C_1(S, B), (1 - \tau) C_2(S, B)), \quad 0 \leq \tau \leq 1, \tag{20}$$

where $C_i(S, B), i = 1, 2$, is given by (4), (8), (10), or (11), depending on the side information availability and the power control strategy. Since time division renders the two users orthogonal, the time-varying power $S_j[i]$ can be optimized for each user independently based on the available side information.

Fixed frequency division divides the total channel bandwidth B into nonoverlapping segments B_1 and B_2, which also reduces the two-user channel to independent single-user channels. The total average power S can be divided between the two users in any way such that their power sum $S_1 + S_2 = S$. For fixed power and bandwidth division, the rate region is then given by

$$(R_1, R_2) = (C(S_1, B_1), C(S_2, B_2)), \tag{21}$$

where, as in the time-division case, $C(S_i, B_i)$ is given by (4), (8), (10), or (11).

When side information is available at the transmitter, the time division rate region will dominate the fixed frequency division rate region over some range of power allocations $(S_1, S_2 : S_1 + S_2 = S)$ in particular when all of the power is allocated to one of the frequency bands (e.g. $S_1 = S, S_2 = 0$). We show in [8] that, as in the time-invariant case, the fixed frequency division rate region intersects the time division line at the point where the power allocation between the two channels is proportional to the bandwidth.

If both the power and the bandwidth partition are allowed to vary for each pair of noise values (n_1, n_2) then the resulting rate region will dominate both fixed frequency division and time division. The achievable rates in this case are given by

$$(R_1, R_2) = \max_{(S_{n_1}, S_{n_2}, B_{n_1}, B_{n_2})} \int (C_{n_1}(S_{n_1}, B_{n_1}), C_{n_2}(S_{n_2}, B_{n_2}))p(n_1, n_2), \tag{22}$$

where $C_{n_i}(S_{n_i}, B_{n_i}) = B_{n_i}\log(1 + S_{n_i}/(n_i B_{n_i}))$, $p(n_1, n_2)$ denotes the joint distribution of n_1 and n_2, $B_{n_1} + B_{n_2} = B$, and the S_{n_i}s satisfy the power constraint $\int (S_{n_1} + S_{n_2})p(n_1, n_2) = S$. Both the power and bandwidth allocations are optimized jointly to achieve the maximum in (22), so the two users are no longer independent. Clearly, any rate point achievable with fixed frequency-division or time-division can also be achieved with this scheme. The maximization in (22) requires a multidimensional optimization subject to the bandwidth and power constraints. The region is also bounded above by the rate region of superposition coding with successive decoding, described below.

The idea of reallocating bandwidth as the channel varies is closely related to dynamic channel allocation, where channel allocation is based on the noise (and interference) levels in a particular frequency band [13]. The frequency allocation of (22) suggests that instead of using a threshold level to determine which user should occupy the channel, the channel should be allocated to the user which gets the most capacity from it.

We now consider superposition coding with successive decoding. The achievable rates in this case are again weighted averages of the achievable rates on each channel:

$$(R_1, R2) = \max_{S_{n_1}, S_{n_2}} \int \left(B\log\left[1 + \frac{S_{n_1}}{n_1 B + S_{n_2} 1[n_1 \geq n_2]}\right], B\log\left[1 + \frac{S_{n_2}}{n_2 B + S_{n_1} 1[n_2 > n_1]}\right]\right) p(n_1, n_2), \tag{23}$$

where $1[\cdot]$ is the indicator function and the S_{n_i}s must satisfy the power constraint $\int (S_{n_1} + S_{n_2})p(n_1, n_2) = S$. Since superposition codes dominate time and frequency division in the time-invariant case, we expect this to be true for time-varying channels as well. Indeed, consider any achievable rate point in the frequency division rate region (22). Associated with that point will be a set of frequency divisions (B_{n_1}, B_{n_2}) and a set of transmit power values (S_{n_1}, S_{n_2}) associated with the noise pair (n_1, n_2). Let $S_{1,2} = S_{n_1} + S_{n_2}$. From §3.1 there exists a superposition code with total power $S_{1,2}$ that dominates the frequency division code on the broadcast channel with noise density values (n_1, n_2). Since we can find such a dominating code for all pairs of noise density values, the weighted integral of the superposition rates over all joint noise density pairs dominates the frequency division rate region of (22).

The rate region for superposition coding without successive decoding is given by (23) with the indicator functions replaced with unity. Since each term in this integral is then convex, the resulting rate region will also be convex. It is therefore dominated by both time- and frequency-division.

To summarize, the time-varying rate region is obtained by taking a weighted average of time-invariant rate regions, with the weights determined by the probability distribution of the noise density pairs. Numerical evaluation of the rate regions defined by (20) and (21) is straightforward using the methods defined in §2. These rate regions have the same general shape as in Figure 4, although they are smaller for the fading channel than for the AWGN channel. Evaluation of (22) and (23) requires multidimensional optimization. These regions can be lower bounded by keeping the transmit power allocated to each user constant. Surprisingly, the resulting rate region is worse than time- or frequency-division over some range of rate pairs. We are currently investigating numerical methods and tighter bounding techniques to evaluate (22) and (23).

4 Conclusions

We've compared the capacity of single-user and broadcast time-varying channels under different spectrum-sharing techniques. We've shown that in the single user case, transmitter adaptation reduces receiver complexity but does not significantly increase capacity. For broadcast fading channels, superposition coding with successive decoding and optimized code adaptation maximizes the capacity region. Frequency-division and time-division have equal rate regions if transmit power is varied according to the changing channel. In addition, dynamic channel (or timeslot) allocation will increase the efficiency of channel use. Finally, superposition coding without successive decoding has the smallest rate region, but also the lowest complexity, since there is no complex decoding or dynamic allocation of power, bandwidth, or timeslots.

References

[1] K. S. Gilhousen, I. M. Jacobs, R. Padovani, A. J. Viterbi, L. A. Weaver, Jr., and C. E. Wheatley III, "On the capacity of a cellular CDMA system," *IEEE Trans. Vehic. Technol.*, Vol. VT-40, No. 2, pp. 303–312, May 1991.

[2] C.-L. I, L.J. Greenstein, and R.D. Gitlin, "A microcell/macrocell cellular architecture for low- and high-mobility wireless users," *IEEE J. Select. Areas Commun.*, Vol. JSAC-11, No. 6, pp. 885–891, Aug. 1993.

[3] A.J. Goldsmith and P.P. Varaiya, "Capacity, mutual information, and coding for finite-state Markov channels," *IEEE Trans. Inform. Theory*, To Appear.

[4] S.-G. Chua and A.J. Goldsmith, "Variable-rate variable-power MQAM for fading channels," *Submitted to the IEEE 1996 Vehic. Tech. Conf.*

[5] J. Wolfowitz, *Coding Theorems of Information Theory.* 2nd Ed. New York: Springer-Verlag, 1964.

[6] T. Cover and J. Thomas, *Elements of Information Theory.* New York: Wiley, 1991.

[7] R.J. McEliece and W. E. Stark, "Channels with block interference," *IEEE Trans. Inform. Theory*, Vol IT-30, No. 1, Jan. 1984.

[8] A. Goldsmith "Design and performance of high-speed communication systems in time-varying radio channels," Ph.D. dissertation, Dept. Elec. Engin. Comput. Science, University of California at Berkeley, 1994.

[9] R.G. Gallager, *Information Theory and Reliable Communication.* New York: Wiley, 1968.

[10] A. Goldsmith and P. Varaiya, " Increasing spectral efficiency through power control," *Proc. IEEE Intl. Commun. Conf.*, pp. 600-604, May 1993.

[11] P.P. Bergmans and T.M. Cover, "Cooperative broadcasting," *IEEE Trans. Inform. Theory*, Vol IT-20, No. 3, pp. 317–324, May 1974.

[12] P.P. Bergmans, "A simple converse for broadcast channels with additive white Gaussian noise," *IEEE Trans. Inform. Theory*, Vol IT-20, No. 2, pp. 279–280, March 1974.

[13] J.C.-I. Chuang, "Performance issues and algorithms for dynamic channel assignment," *IEEE J. Selected Areas Commun.*, Vol. SAC-11, No. 6, pp. 955–963, Aug. 1993.

Resource Allocation for Packet Data Transmission in DS-CDMA

Michael L. Honig and Joon Bae Kim

Department of EECS, Northwestern University
2145 Sheridan Road, Evanston, IL 60208

ABSTRACT

We consider a packet data Direct-Sequence-Code-Division Multiple-Access system with different classes of data users. Each class is associated with a particular Quality of Service (throughput and delay), and each user within a class generates fixed-length packets at the same rate. Reliable communications is achieved through error detection and retransmission (ARQ). A different QoS for each class of users can be achieved by varying the power and/or processing gain assigned to each class. For a single class of users we compute the processing gain that maximizes average throughput as a function of the probability of retransmission parameters (which depends on the coding scheme), number of users, packet length, and transmitted power. The optimal processing gain increases linearly with the number of users (assuming the other parameters are fixed). We then consider two classes of users and jointly assign powers and processing gains to the two classes to optimize the tradeoff between the number of users in each class, subject to QoS constraints. Numerical results are presented that illustrate this tradeoff, assuming a particular error detection scheme with stop-and-wait ARQ.

1. Introduction

Direct-Sequence-Code-Division Multiple-Access (DS-CDMA) is a strong contender for wireless access in applications such as digital mobile cellular and Personal Communications Services (PCS). The current IS-95 air interface for DS-CDMA is designed to handle voice traffic. However, future multimedia wireless services may include many different traffic streams consisting of voice, data, and perhaps images or video. Each data stream is likely to have different requirements on error rate, delay, and delay jitter. The general problem we wish to address is how to maximize the number of active users subject to performance constraints in the context of DS-CDMA. In this paper, we consider only data transmission.

There are many ways in which the performance characteristics of different traffic streams can be controlled in DS-CDMA. For example, to decrease the error rate associated with a particular user, the transmitted power can be increased. However, this increases interference to other users, degrading their performance. Another possibility is to increase the processing gain by increasing the chip rate and/or the symbol duration. However, increasing the chip rate requires additional bandwidth, and increasing the symbol duration lowers the transmitted symbol rate. Other ways to achieve different performance characteristics for different data streams, which we do not consider here, are to use different coding/decoding and detection schemes that vary in complexity.

We consider a multiuser packet data DS-CDMA system in which each user generates a sequence of fixed-length packets, and reliable communication is achieved through error detection and retransmission (ARQ). In this work, we consider only Stop-and-Wait Automatic Repeat Request (SW-ARQ); however, our results can be easily extended (at least conceptually) to Go-Back-N and Selective-Repeat ARQ [1].

We assume that each user spreads across the total available bandwidth by using the same chip rate. This maximizes the amount of spreading, and therefore maximizes multipath resolution and resistance to interference. Different performance, or Quality of Service (QoS), is achieved for different users by varying the transmitted powers and processing gains. The processing gain is varied by changing the symbol duration. That is, doubling the processing gain doubles the symbol duration. Note that decreasing the symbol rate can actually increase throughput. This is because increasing the processing gain increases the received Signal-to-Interference plus Noise-Ratio (SINR), and therefore decreases the probability of retransmission.

The users are divided into different *classes* which have different performance requirements, and may also have different traffic characteristics (i.e., packet generation rates, packet lengths). Within each class of users, the traffic characteristics and performance requirements are assumed to be the same. We also assume that the *same* transmitted power and the *same* processing gain is assigned to *all* users within each class. The performance measures we consider here are average throughput, and the probability that the delay exceeds a given threshold. Related work on supporting multi-rate traffic with different QoS requirements in the context of DS-CDMA is presented in [2]-[9]. The approach taken here differs from previous work in that we consider how to allocate both transmitted powers and processing gains to maximize system capacity, assuming different classes of data users.

We start by considering a single class of users, and studying the preceding performance measures as a function of processing gain, keeping the transmitted power fixed. The processing gain that maximizes throughput is characterized as a function of the number of users, packet length, transmitted power, and probability of retransmission parameters, which depends on the coding scheme. We find that for powerful coding schemes, the throughput decreases rapidly when the processing gain decreases below the optimal point. However, when the processing gain is increased above the optimal point, throughput decreases gradually. These results assume perfect power control with matched-filter detection.

We then consider two classes of users, and solve for the capacity region, subject to constraints on the probability that the delay exceeds a threshold. We also consider the problem of jointly selecting the transmitted powers and processing gains assigned to both classes to maximize the throughput of one class, given a constraint on the throughput of the other class. The solution is used to plot a feasibility region which shows what combinations of throughput are possible for the two classes, given the number of users in each class, and the error control scheme.

2. System Model

We consider the reverse link of a cellular system. There are C classes of asynchronous data users within a single cell, and class i contains K_i users, $1 \leq i \leq C$. Initially, we consider only one class of users, and subsequently expand to two classes. Within class i, each active user generates a sequence of fixed-length packets at the same average rate λ_i. The packet length for users in class i is denoted as L_i. All users transmit at a constant chip rate (roughly the available bandwidth).

Assuming matched-filter detection, the QoS for user i is a function of the SINR, defined as [10]

$$\mathrm{SINR}_i = \frac{P_i}{\frac{\alpha}{N_i}\sum_{k\neq i}^{K} g_k \beta_k P_k + \sigma^2} \tag{1}$$

where P_i is the transmitted energy per bit, and N_i is the processing gain, both for class i, α is a constant that depends on the shape of the DS-CDMA chips [11] (we will assume $\alpha = 1/3$), g_k is the attenuation from transmitter k to the base station relative to the desired signal, β_k is an on/off indicator for user k, $K = \sum_{i=1}^{C} K_i$ is the total number of users, and σ^2 is the background noise variance which may include inter-cell interference. In this paper, we assume perfect power control (which may assign different powers to the users), so that $g_k = 1$ for all k. The on/off indicators β_k are assumed to be independent random variables with distribution

$$\mathrm{Prob}[\beta_k = 1] = p_{\beta_i} \qquad \mathrm{Prob}[\beta_k = 0] = 1 - p_{\beta_i} \tag{2}$$

where i denotes the class. Consequently, SINR_i is a random variable with distribution that can be computed given the parameters p_{β_i}, $i = 1, \cdots, C$. If user k in class i generates packets according to a Poisson process, and buffers all packets until acknowledgment is received, then it is possible to compute p_{β_i} from the arrival rate λ_i [4]. However, we will directly assign values to p_{β_i}, and not consider the relationship between λ_i and p_{β_i}.

3. Retransmission Strategy and Performance Measures

We assume Stop-and-Wait Automatic Repeat Request (SW-ARQ) to ensure reliable data communications. To simplify the analysis, acknowledgements are always assumed to be correctly received. This is a reasonable assumption provided that a powerful coding scheme is used over the feedback channel. The minimum transmission time of a single packet for a user in class i can then be expressed as [1, 12]

$$T_{min_i} = \frac{L_i N_i}{R_c} + T_d \tag{3}$$

where L_i is the number of bits per packet, N_i is the processing gain, both for class i, R_c is the chip rate, and T_d is the sum of the propagation time and the processing time which is required to detect and/or correct the errors in a packet. A processing time might be on the order of 1 *ms* [1], and the propagation time is assumed to be negligible. Here, T_d is assumed to be a constant (3 *ms*) for every transmission.

Let T_i denote the transmission time of a single packet for a user in class i, which consists of T_{min_i} plus a random delay Δ due to retransmissions. T_i has the geometric distribution

$$\mathrm{Prob}[\mathrm{T}_i = j \cdot T_{min_i}] = \bar{\mathrm{p}}_{r_i}^{\,j-1}(1 - \bar{\mathrm{p}}_{r_i}) \tag{4}$$

where $j = 1, 2, 3, \cdots$ is the number of transmissions necessary for successful reception and $\bar{\mathrm{p}}_{r_i}$ is the probability of retransmission, which must be averaged over the distribution for SINR_i. The probability of retransmission p_{r_i} is a random variable since SINR_i is a random variable. The average packet transmission time is

$$\mathrm{E}[\mathrm{T}_i] = \frac{T_{min_i}}{1 - \bar{\mathrm{p}}_{r_i}} \tag{5}$$

and the average throughput is defined as

$$\eta_i = \frac{l_i}{\mathrm{E}[\mathrm{T}_i]} \tag{6}$$

where l_i is the total number of information bits in a packet. Since l_i is fixed, $\mathrm{E}[\mathrm{T}_i]$ and η_i are basically the same measure of performance. In addition to $\mathrm{E}[\mathrm{T}_i]$ and η_i, we introduce another measure of performance q_i, which is the probability that the packet transmission time exceeds a given threshold T_{max_i}, the maximum acceptable packet transmission time. From (4) we have that

$$q_i = \mathrm{Prob}[\mathrm{T}_i > T_{max_i}] = \bar{\mathrm{p}}_{r_i}^{m} \quad \text{where} \quad m = \lfloor (T_{max_i}/T_{min_i}) \rfloor \tag{7}$$

and $\lfloor x \rfloor$ is the largest integer less than or equal to x. Note that $q_i < 1$ implies that $T_{min_i} < T_{max_i}$, which from (3) implies an upper limit on the processing gain. Namely, the maximum processing gain for a given T_{max_i} is

$$N_{max_i} = \lfloor (T_{max_i} - T_d) R_c / L_i \rfloor \tag{8}$$

We consider both $\mathrm{E}[\mathrm{T}_i]$ (or η_i) and q_i as QoS measures. These depend on the user's processing gain N_i and retransmission probability p_{r_i}. Moreover, p_{r_i} depends on the coding scheme and the channel symbol error (crossover) probability, which is a function of the modulation scheme and the user's SINR. In what follows we approximate

$$\mathrm{p}_{r_i} \approx a \exp\left[-b \cdot \mathrm{SINR}_i\right] \tag{9}$$

where a and b are constants.[1] The retransmission probability exponent b therefore characterizes the strength of the coding scheme. Note that $\bar{\mathrm{p}}_{r_i}$ in (5) is p_{r_i} averaged over the distribution for SINR_i.

For the numerical results, we consider the following two coding schemes:

1. (24,12,8) extended Golay code with 40 codewords per packet
2. (60,48,8) Reed-Solomon outer code with a rate 1/2 convolutional inner code

Scheme 1 is a relatively simple error detecting code for use with pure ARQ [12], and scheme 2 is a powerful coding scheme for a type-I hybrid ARQ protocol, which was proposed in [13]. In scheme 1 there are 960 bits per packet of which 480 bits are information bits including the packet sequence number (code rate = 0.5). In scheme 2 there are 960 bits in a packet of which 384 bits are information bits including the packet sequence number (code rate = 0.4). In both cases we assume perfect interleaving so that successive symbol errors are independent events.

The retransmission probability for each of the preceding schemes can be computed given the results in [12] and [13]. In both cases we assume BPSK modulation. However, for the first scheme we assume an additive Gaussian noise channel, and for the second scheme we assume nonselective Rayleigh fading. In both cases the approximation (9) is accurate for $\mathrm{p}_{r_i} < 10^{-1}$. For scheme 1, $a = 114.04$ and $b = 0.515$, and for scheme 2 $a = 7.09 \times 10^{16}$ and $b = 16.92$. The large difference in error exponent b for the two schemes leads to significant differences in the numerical results.

---

[1] Depending on the coding scheme, approximating p_{r_i} by (9) where SINR_i is specified in dB may be more accurate. In this case, p_{r_i} can be approximated as $a(\mathrm{SINR}_i)^{-b}$.

4. Single Class of Users

We start by assuming only one class of users, so that all users transmit with the same power P and processing gain N. In this section we drop the subscript which denotes class. From (1), we have

$$\mathrm{SINR} = \frac{P}{\frac{\alpha}{N}\mathrm{B}P + \sigma^2} \qquad \text{where} \quad \mathrm{B} = \sum_{k=1}^{K-1} \beta_k \tag{10}$$

for all users. (2) and (10) imply that B has the binomial distribution

$$\mathrm{Prob}[\mathrm{B} = j] = \binom{K-1}{j} p_\beta^j (1 - p_\beta)^{K-1-j} \tag{11}$$

where $j = 0, 1, \cdots, K-1$, and $\bar{\mathrm{B}} = \mathrm{E}[\mathrm{B}] = p_\beta(K-1)$. Jensen's inequality implies

$$\mathrm{E}[\mathrm{SINR}] \geq \frac{P}{\frac{\alpha}{N}\bar{\mathrm{B}}P + \sigma^2} = \frac{PN}{\alpha p_\beta (K-1)P + \sigma^2 N} = \mathrm{SINR}(\bar{\mathrm{B}}) \tag{12}$$

Fig. 1 shows plots of E[SINR] and SINR($\bar{\mathrm{B}}$) vs. N and P when $K = 20$, $p_\beta = 0.9$, and $\sigma^2 = 1\ mW$. In general, for this σ^2 E[SINR] is accurately approximated by SINR($\bar{\mathrm{B}}$) if $K > 5$. In addition, to simplify the following analysis, we will use the approximation

$$\bar{\mathrm{p}}_r \approx a \exp\left[-b \cdot \mathrm{SINR}(\bar{\mathrm{B}})\right] \tag{13}$$

which we found to be fairly accurate for $\mathrm{p}_r < 10^{-1}$.

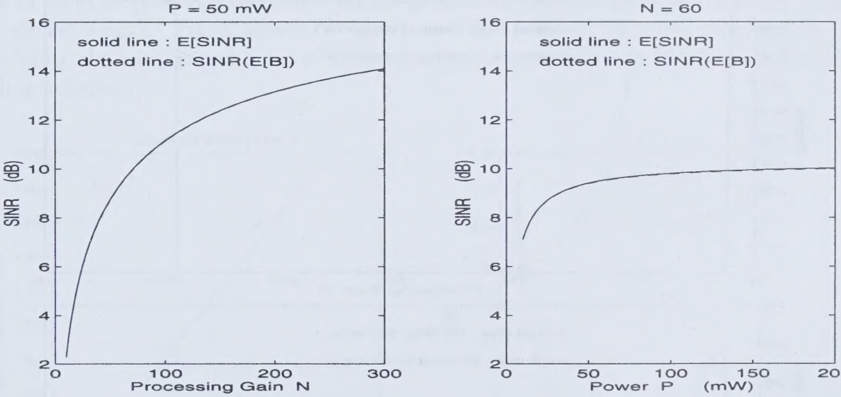


Fig. 1: E[SINR] vs. SINR($\bar{\mathrm{B}}$) with different P and N

Fig. 1 and (12) both indicate that increasing the power P does not have much impact on SINR (and also on $\bar{\mathrm{p}}_r$, E[T], and q) for small levels of background noise σ^2. However, increasing the processing gain N increases the SINR, which decreases the retransmission probability. There is therefore a tradeoff between the decrease in E[T] caused by this decrease in $\bar{\mathrm{p}}_r$, and the increase in E[T] caused by the increase in symbol duration. The decrease in E[T] predominates for small N, and the increase in E[T] predominates for large N. To find the processing gain N that minimizes E[T], we take the derivative of (5) with respect to N, using the approximation (13). The optimum processing gain N^* must satisfy

$$\exp\left[\frac{bPN^*}{\alpha p_\beta(K-1)P+\sigma^2 N^*}\right] = a\left[1+\left(N^*+\frac{T_d R_c}{L}\right)\frac{\alpha p_\beta(K-1)bP^2}{\left(\alpha p_\beta(K-1)P+\sigma^2 N^*\right)^2}\right] \tag{14}$$

Plots of E[T], q, and η vs. processing gain N for the two coding schemes considered are shown in Fig. 2. In both cases, the number of users K is 20, the power P is 50 *mW*, the chip rate R_c is 5 *MHz*, and the background noise variance σ^2 is 1 *mW* so that the SINR is approximately 17 *dB* in the absence of interference. T_{max} is arbitrarily taken to be 50 *ms*. These plots do *not* assume the approximations used to obtain (14). Solving (14) for N^* in this case gives 98 for conding scheme 1, and 16 for coding scheme 2 which corresponds to the minima shown in Fig. 2 (99 and 15 respectively).

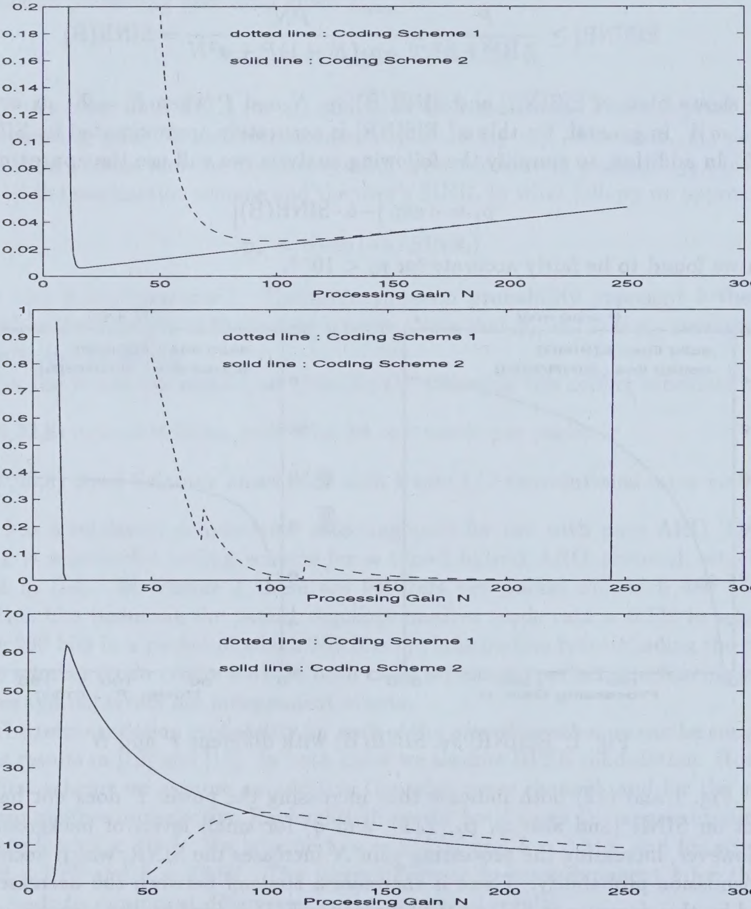


Fig. 2: E[T], q, and η vs. N for two different coding schemes

Fig. 2 shows that the average throughput η decreases rapidly when the processing gain N decreases below the optimal point N^*. This happens because the SINR decreases approximately linearly, which causes the retransmission probability $\bar{\mathrm{p}}_r$ to approach one. However, when N increases above N^*, η decreases relatively slowly. This is because $\bar{\mathrm{p}}_r$ approaches zero, so that E[T] approaches the time it takes to transmit a single packet (T_{min}), which increases linearly with the symbol time (N/R_c). The decrease in η for $N < N^*$ is more severe for the more powerful coding scheme since for a more powerful code, small changes in the SINR cause more dramatic changes in $\bar{\mathrm{p}}_r$. These results indicate that the choice of processing gain can significantly affect throughput.

The discontinuity in q shown in Fig. 2 is due to the fact that the packet transmission time (including retransmissions) assumes *discrete* values which are multiples of the time it takes to transmit the packet without retransmission. Consequently, an incremental change in processing gain can change the number of retransmissions needed to cause the packet transmission time to exceed T_{max}. Depending on the application, which determines T_{max}, it is possible that the N which minimizes average delay results in an unacceptably high value of q. In general, a reasonable choice for N is the minimum N for which $q < \xi$ for some threshold probability ξ. Combining the expression for SINR (10) with (7) and the approximation (13), we find that given a particular N (i.e., $N = N^*$) and $0 < \xi < 1$, to ensure $q < \xi$ we must have

$$T_{max} \geq \left(\frac{LN^*}{R_c} + T_d\right) \frac{\ln \xi}{\ln a - \frac{bPN^*}{\alpha p_\beta (K-1)P + \sigma^2 N^*}} \qquad (15)$$

By taking the derivative of both sides of (14) with respect to K, it can be shown that the optimal processing gain increases nearly linearly with the number of users. The rate of increase (dN^*/dK) depends on the probability of error parameters a and b, and the packet length. Fig. 3 shows N^* vs. K and dN^*/dK vs. the error exponent b. In this figure the background noise σ^2 is assumed to be zero, and a is 114.04 (based on the coding scheme 1).

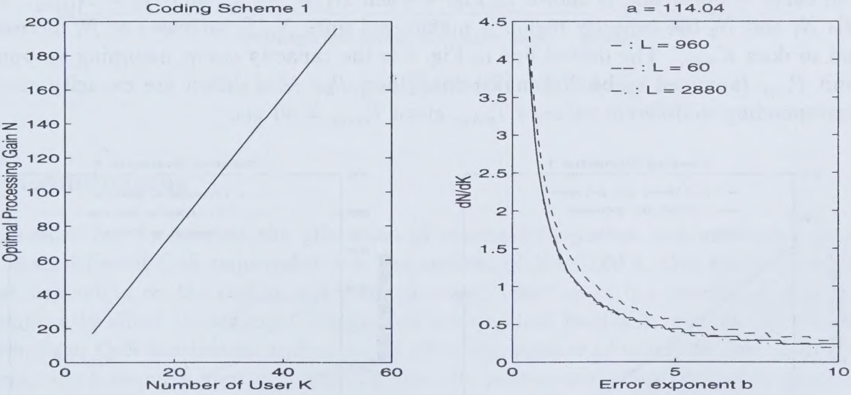


Fig. 3: N^* vs. K, and dN^*/dK vs. error exponent b

5. Two Classes of Users

We now consider the case where there are two classes of data users in a cell. We assume the same coding scheme and packet length L for all users. The averaged SINR for users in each class is then given approximately by

$$\begin{aligned} \mathrm{E}[\mathrm{SINR}_1] &\approx \frac{P_1 N_1}{\alpha p_{\beta_1}(K_1-1)P_1 + \alpha p_{\beta_2} K_2 P_2 + \sigma^2 N_1} \\ \mathrm{E}[\mathrm{SINR}_2] &\approx \frac{P_2 N_2}{\alpha p_{\beta_1} K_1 P_1 + \alpha p_{\beta_2}(K_2-1)P_2 + \sigma^2 N_2} \end{aligned} \tag{16}$$

Unlike the single-class case, the power assignment (P_1 and P_2) can change $\mathrm{E}[\mathrm{SINR}_i]$ significantly even if there is no background noise σ^2. Moreover, increasing P_1 not only improves $\mathrm{E}[\mathrm{SINR}_1]$ but also degrades $\mathrm{E}[\mathrm{SINR}_2]$. On the other hand, changing the processing gain N_1 only affects $\mathrm{E}[\mathrm{SINR}_1]$, and *not* $\mathrm{E}[\mathrm{SINR}_2]$.

We now compute the system capacity, which is the maximum number of class 1 and class 2 users that can be accommodated subject to constraints on QoS (7). That is, we want to maximize the number of class 2 users K_2, given K_1, and subject to $q_1 < \xi_1$ and $q_2 < \xi_2$. For each class i, given a fixed maximum transmission time T_{max_i}, processing gain N_i, and threshold probability ξ_i, the minimum required $\mathrm{E}[\mathrm{SINR}_i]$ is

$$\mathrm{E}[\mathrm{SINR}_i] > S_i \qquad \text{where} \quad S_i = \frac{1}{b}\ln\left[a(\xi_i)^{-1/m_i}\right] \tag{17}$$

This allows us to obtain combinations of K_1 and K_2 for which there exists feasible power assignments P_1 and P_2. If we assume that there is no power limit, then the capacity curve is specified by

$$\text{slope} = -\frac{S_1 p_{\beta_1}(N_2 + S_2\alpha p_{\beta_2})}{S_2 p_{\beta_2}(N_1 + S_1 \alpha p_{\beta_1})}, \quad K_{max_1} = \frac{N_1}{S_1 \alpha p_{\beta_1}} + 1, \quad K_{max_2} = \frac{N_2}{S_2 \alpha p_{\beta_2}} + 1 \tag{18}$$

The curve is linear and is shown in Fig. 4 when $N_1 = N_{max_1}$, and $N_2 = N_{max_2}$. For this N_1 and N_2 the capacity region is maximized since N_i/S_i increases as N_i increases and so does K_{max_i}. The dotted line in Fig. 4 is the capacity curve, assuming an upper limit P_{max} (assumed to be 200 *mW*) on P_1 and P_2. Also shown are capacity curves corresponding to different value of T_{max_2} given $T_{max_1} = 50$ *ms.*

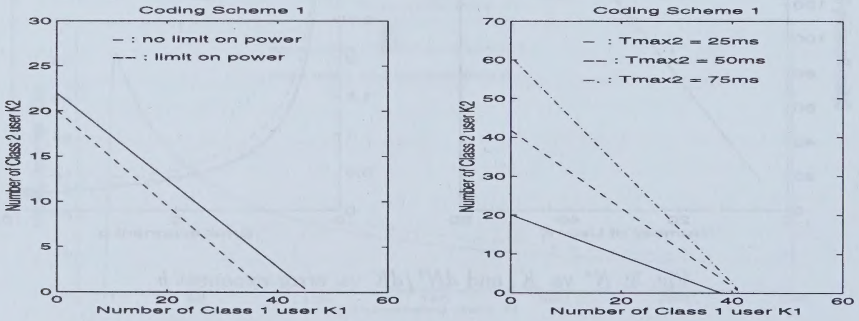


Fig. 4: K_1 vs. K_2 with different value of T_{max_2}

Finally, given a fixed number of users in each class, we consider the problem of assigning powers and processing gains to maximize throughput. That is, given K_1, K_2, and η_1, we wish to maximize η_2. More precisely, our problem is to

$$\text{minimize } \mathrm{E}[\mathrm{T}_2] \text{ with respect to } P_1,\ P_2 \text{ and } N_2 \text{ subject to } \mathrm{E}[\mathrm{T}_1] = \tau_1.$$

From (16), for fixed N_1 and N_2, $\mathrm{E}[\mathrm{SINR}_2]$ is an increasing function of P_2 and a decreasing function of P_1. Therefore, in the absence of background noise the preceding problem can be solved in two steps:

1. max P_2/P_1 subject to the constraint $\min_{N_1} \mathrm{E}[\mathrm{T}_1] = \tau_1$.
2. min $\mathrm{E}[\mathrm{T}_2]$ over N_2 (for fixed P_2/P_1)

Throughput regions computed via the preceding algorithm are shown in Fig. 5. Note that these depend mainly on the number of class 1 and class 2 users, the packet length L, and the probability of retransmission parameters a and b.

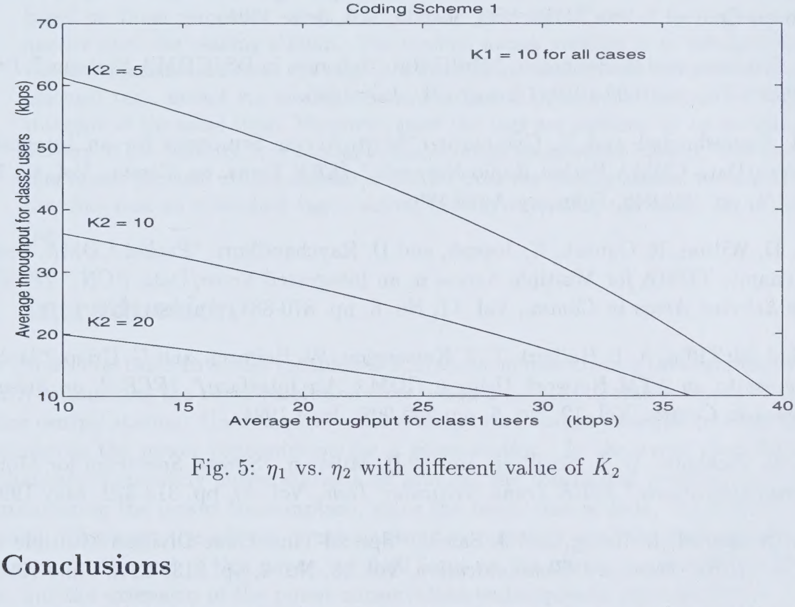


Fig. 5: η_1 vs. η_2 with different value of K_2

6. Conclusions

This work has focused on the allocation of transmitted powers and processing gains to meet different QoS requirements in the context of DS-CDMA. Our results indicate that depending on the coding and ARQ strategy, selection of the processing gain can significantly affect throughput. We have also computed feasibility regions (K_1 vs. K_2 given fixed QoS constraints and η_1 vs. η_2 given the number of users) for two classes of users, which requires joint optimization over the powers and processing gains assigned to the two classes. We remark that another way to achieve multiple information rates, which we have not considered, is to assign multiple codes to the different classes of users [3, 9]. Other possibilities for future work include consideration of more than two classes of users, voice traffic in addition to the data classes, and impairments such as fading, imperfect power control, and other-cell interference.

References

[1] J. Walrand, *Communication Networks : A First Course*, Aksen Associates, Inc., Boston, MA, 1991.

[2] C-L I and K. K. Sabnani, "Variable Spreading Gain CDMA with Adaptive Control for Integrated Traffic in Wireless Networks," *Proc. 1995 VTC*, pp. 794-798, Chicago, IL, July 1995.

[3] C-L I, G. P. Pollini, L. Ozarow, and R. D. Gitlin, "Performance of Multi-Code CDMA Wireless Personal Communications Networks," *Proc. 1995 VTC*, pp. 907-911, Chicago, IL, July 1995.

[4] E. Geraniotis, Y.-W. Chang, and W.-B. Yang, "Dynamic Code Allocation for Integrated Voice and Multi-Priority Data Traffic in CDMA Networks," *European Trans. on Telecomm.*, Vol. 6, No. 1, pp. 85-96, January-February 1995.

[5] L. C. Yun and D. G. Messerschmitt, "Variable QOS in CDMA Systems by Statistical Power Control," *Proc. 1995 ICC*, Seattle, WA, June 1995.

[6] T. Ottosson and A. Svensson, "Multi-Rate Schemes in DS/CDMA Systems," *Proc. 1995 VTC*, pp. 1006-1010, Chicago, IL, July 1995.

[7] M. Soroushnejad and E. Geraniotis, "Multi-Access Strategies for an Integrated Voice/Data CDMA Packet Radio Network," *IEEE Trans. on Comm.*, Vol. 43, No. 2/3/4, pp. 934-945, February-April 1995.

[8] N. D. Wilson, R. Ganesh, K. Joseph, and D. Raychaudhuri, "Packet CDMA Versus Dynamic TDMA for Multiple Access in an Integrated Voice/Data PCN," *IEEE J. on Selected Areas in Comm.*, Vol. 11, No. 6, pp. 870-884, August 1993.

[9] M. J. McTiffin, A. P. Hulbert, T. J. Ketseoglou, W. Heimsch, and G. Crisp, "Mobile Access to an ATM Network Using a CDMA Air Interface," *IEEE J. on Selected Areas in Comm.*, Vol. 12, No. 5, pp. 900-908, June 1994.

[10] R. L. Pickholtz, D. L. Schilling, and L. B. Milstein, "Spread Spectrum for Mobile Communications," *IEEE Trans. Vehicular Tech.*, Vol. 40, pp. 313-322, May 1991.

[11] P. Crespo, M. L. Honig, and J. Salehi, "Spread-Time Code-Division Multiple Access," *IEEE Trans. on Communication*, Vol. 43, No. 6, pp. 2139-2148 June 1995.

[12] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1995.

[13] R. D. Cideciyan and E. Eleftheriou, "Concatenated Coding Scheme for Reliable Data Transmission in CDMA Cellular Systems," *Int'l Zurich Seminar on Digital Communications*, 1994.

Algorithms for Power-Efficient Identification of Radio-Frequency Identification Tags

Arvind Krishna and Dilip D. Kandlur
IBM Thomas J. Watson Research Center
Yorktown Heights, NY 10598
`krishna,kandlur@watson.ibm.com`

Abstract

This paper describes a method to minimize power consumption for an identification process. The goal is to identify all items present within the vicinity of a reading station. Items have tags on them, each tag consisting of electronic circuitry activated via a radio frequency (wireless) signal from the reading station. The identity of the tags is not known *a priori* to the reading station. Tags respond to queries from the reading station, and the reading station can identify the tags based on these responses. Tags are normally powered off except in response to queries from the reading station. The random access problem is to schedule the various transmissions from the tags in a distributed fashion with no coordination amongst tags, except via messages from the base station, such that no two tags transmit at the same time. Moreover, since the tags are powered by an on-board battery of low capacity, it is critical to conserve the usage of the battery. Our solution to this problem utilizes a series of queries from the reading station to minimize the time that an individual tag is active, thereby extending the useful life of the tag.

1 Introduction

The traditional random access techniques[1, 2] focus on maximizing throughput, or equivalently minimizing the batch resolution time, when a number of stations need to transmit to one central station. Unfortunately, these techniques do not promise, or even attempt, to minimize the power consumption for a given station. In the event that there is no mechanism to remotely power stations off and on, the traditional problem is equivalent to minimizing the power consumption, since the faster one is done, the less power may be used. However, the advent of mechanisms to remotely activate stations allows new solutions. In the rest of this paper, we shall focus on the particular case of identification tags, but the extension of the power conservation techniques to other situations, such as tele-metering or data collection, is straightforward.

Low-cost electronic circuitry and FCC regulations enable us to envision a future in which a small electronic **tag** [3, 4, 5, 6, 7, 8] is stuck on items. This tag carries information about the item, such as its identity, price, date on shelf, and so on (much like a bar code). This tag carries an electronic circuit for receiving radio signals and digital control circuitry [9, 10]. The reception of an appropriate radio signal triggers the tag to power on its circuitry and respond to further messages received over radio signals. A number of research laboratories and corporations around the globe have recently started programs to investigate and/or prototype such tags, including Amtech Corp., AT&T Corp., IBM Corp., Sensormatic Industries, Hughes Electronics, the CSIR government laboratories in South Africa, David Sarnoff Laboratories, etc. We describe a few typical situations to motivate the problem. The first case is for inventory control and pricing. A shopper loads a cart with items, each item having an identification tag, approaches a tag reader in a checkout lane which queries the tags and determines the prices and flashes the total (and bill) to the shopper who then uses a smart card to pay the bill and proceeds outside, resulting in a no-wait checkout. Extensions of this example for pricing and inventory control are straightforward, assuming that the tag can be written as well as read. A second example would be airline passengers, each having a smart card with such an identification tag. After the passengers are seated in an airplane, a tag reader could query the tags and if the responses match the number of seated passengers and the presumed identities of the passengers in the airline computer, then no manual check-in is required.

To preserve the bar code paradigm, these tags have a similar form factor, albeit a little thicker. This form factor precludes a replaceable battery and thus the tags are powered by an on-board battery, formed of appropriate polymer technology with just a few layers of foil put together like a sandwich. It is of paramount importance to conserve the usage of this battery, since it is the prime determinant of the useful life of a tag. This battery is consumed primarily when the tag is active, as in the case when the tag interacts with the reading station. The identity of the tags is not known *a priori* to the tag reader. For simplicity, we shall assume that our problem is to find the identities of the (multiple) tags in the vicinity of a tag reader. The reading station can initiate the reading process by sending some query that activates all tags within its range. Then, the tags can respond to the received query and the reading station can identify the tags based on these responses. This paper describes algorithms to identify various tags present within the vicinity of a tag reading station. Our methods are unique [11] in that they minimize the battery usage required to identify the tags. Typical random access algorithms would take time proportional to the number of tags to complete the identification. However, we will show that with our methods, on average, each tag has to be activated for a time proportional to the square root of the number of tags, a substantial saving in many situations.

There has been much work done on related topics on power consumption, such as extending the talk time of cellular telephones[12], reducing the power consumption of radio circuits [13], reducing the number of transmissions made by a mobile user[14], reducing the time that a mobile user has to listen to channels by using the concept of multiple paging channels [15], and the traditional work in communication theory to maximize the link quality for a given transmission power. Our work differs from these in certain key aspects. First, the circuit and modulation techniques can be applied in addition to the algorithms that we propose to further reduce the power consumption while a tag is active. Next, since the tags are only required to be identified once (in a short time span) the techniques on mobility management and paging are not relevant. Our goal is to deactivate the tags as much as possible, even during the identification process, unless the tag is actually transmitting since even listening to the channel consumes power.

The rest of this paper is organized as follows. In Section 2, we briefly describe the relevant aspects of the system model. Next, in Section 3 we describe the algorithm to identify the tags when there is only one signal from the reading station that activates all tags within its range. We then show that this algorithm does the best possible. We conclude the section by discussing some issues related to the initial estimate of the number of tags and and some online versions of tag identity resolution. In Section 4, we consider a case where the tags have the capability of being activated only in response to a predetermined signal (amongst multiple signals) with the tags remaining deactivated in the presence of other signals from the reader station. Further, each active tag has the capability of selecting the signal whereby it will next be activated, and each tag can deactivate itself. We conclude the paper by discussing some open problems and expressing some conjectures.

2 The Model

There is one tag reader and N tags within its vicinity. The identities of the tags are not known a priori to the reader and the problem is for the tag reader to resolve the identities of all the tags, i.e., a batch resolution problem. Of course, the packet containing the identity of each tag could also contain information such as its price, last date of inventory, etc. Tags are normally inactive, i.e., powered off and not consuming any power. They are made active in response to a known signal received from the tag reader, e.g., a short high-power radio frequency pulse. Upon activation they listen for transmissions from the tag reader and respond when necessary until they are asked to power off. Logically, we can think of this as a mechanism to make tags active or inactive. Due to the broadcast nature of the activation pulse when one tag is made active, all tags are made active. Thus, following such a *wakeup* message, all tags except those that should be active, have to be told to become inactive or have to deduce this fact from information kept at the tag or transmitted by the tag reader.

Once the tags are activated, there is an RF communication channel, consisting of a forward channel and a reverse channel, between the tag readers and the tags. Transmissions from the base station (the forward channel) do not need scheduling, since the tag reader is the only station to transmit on this channel. However, the reverse channel needs a contention resolution mechanism since all the tags can potentially transmit in this channel. Logically, the channel can be assumed to be time-slotted with tag reader transmissions alternating with tag transmissions. There are several restrictions on the nature of the contention resolution protocol that arise due to the characteristics of the tags. The tags cannot detect collisions, i.e., distinguish that two (or more) of them are transmitting unless the tag reader informs them of this fact, hence early collision detection is not possible. The tag reader must detect collisions (by examining the checksum on the received packet) and inform the tags in the next time-slot. Hence, carrier sensing based contention resolution is not allowed, and the only feedback available to the tags is from the tag reader. Further, the multiplicity of a collision is unknown, the reader only knowing that two or more tags attempted to transmit in a give slot.

As a side remark, each tag has its own local clock which may have a certain drift from the other tags, so it is not possible to bit-synchronize the operations of the different tags. It is therefore desirable to develop an algorithm that is *asynchronous* and does not depend on strict timing information for transmission.

For convenience, and without loss of generality, we assume that a tag consumes a unit of power for two time-slots that it remains on, i.e., a unit of power is consumed for the potential of receiving one packet from the tag reader and transmitting a packet in return. However, a tag remains on for a minimum of β units in response to an activation signal received from the tag reader.

A basic building block for the algorithms that will follow is an algorithm to resolve the identities of a set of active tags. We shall describe such an algorithm here since it will be used in the sequel. In the absence of carrier-sensing, our choice is restricted to either slotted Aloha or splitting algorithms [1, Ch. 4]. Note that reservation based algorithms are not appropriate here, since the reservation request would carry the tag identity and there is no further data to transmit. Since splitting algorithms can provide better performance than slotted Aloha, we will use them and specifically we use a modified binary tree variant of the splitting protocols as described by [16]. Assuming that one transmission by a tag and feedback from the tag reader takes one unit of time, this protocol would take, on average, $n/0.468$ units of time to resolve the identities of n tags. We shall now compute the average time that it takes before a marked tag is resolved. Since we know that the total time to resolve the identities of n tags is proportional to n, we intuitively expect that a given tag is resolved in about half that time, on average. The analysis below will confirm that intuition. Our analysis in what follows is similar to the analysis in [16, 2] (with the difference that our estimate of the number of tags whose identity has to be resolved is perfectly known).

The protocol works by dividing the n active tags in m batches, each tag choosing a particular batch with probability $1/m$. Each batch, of average size n/m is then resolved, in turn. Let $L(n_i)$ denote the time that it takes to resolve a batch of size n_i, and $T(n_i)$ the time that it takes to resolve a given tag in a batch of size n_i. Further, let n_j be the number of tags in batch j, excluding the marked tag. Then, since the marked tag may choose each batch with probability $1/m$, the average time it takes to resolve the marked tag in a batch of n tags, $P(n)$, is given by

$$P(n) = \mathrm{E}\left[\sum_{i=1}^{m}\frac{1}{m}\left(T(n_i+1)+\sum_{j=1}^{i-1}L(n_j)\right).\right] \tag{1}$$

There is a constant α ($\alpha = 2.623$ for the modified binary tree algorithm) such that the expected value of $L(n)$ is less than $\alpha n - 1$ for $n \geq 4$ and $L(0) = L(1) = 1$, $L(2) = 4.4143$, $L(3) = 6.8847$. Further, note that $T(n_i) \leq L(n_i)$. Together with the observation that $\Pr[n_j = k] = \binom{n-1}{k}\left(\frac{1}{m}\right)^k\left(1-\frac{1}{m}\right)^{n-1-k}$, we can emulate the analysis of [2, Thm. 3]. The first step is to observe that Equation (1) implies that $P(n) = \mathrm{E}[T(n_i+1)] + \frac{m-1}{2}\mathrm{E}[L(n_j)]$, immediately allowing us to realize that for large n and m, the analysis in [2, Thm. 3] implies that we can write Eq. (1) as $P(n) \leq \frac{1}{2}2.135n$, where it was assumed that $m \approx 0.786n$. This confirms the earlier intuition. Note that choosing m as a nonlinear function of n will allow us to do a little better in practice when n is small. Numerical experiments confirm the above analysis (for small values of n) and actually show that $P(n)/n$ is monotonically decreasing with n and some sample values are $P(10) \approx 1.162n$, $P(100) \approx 1.077n$ $P(1000) \approx 1.068n$. From now on, we shall assume that there is a universal constant γ such that $P(n) \approx \gamma n$ and that $\gamma \leq 1.0675$.

3 Single-Wakeup Signal

In this section, we describe the basic algorithm to identify the tags when there is only one signal from the reading station that activates all N tags within its range. The flowchart, Fig. 1, describes the basic algorithm where the only parameter left undefined is the number of rounds, k. Let us begin by saying that we choose $k = \lfloor\sqrt{\frac{\gamma N}{\beta}}\rfloor$. The

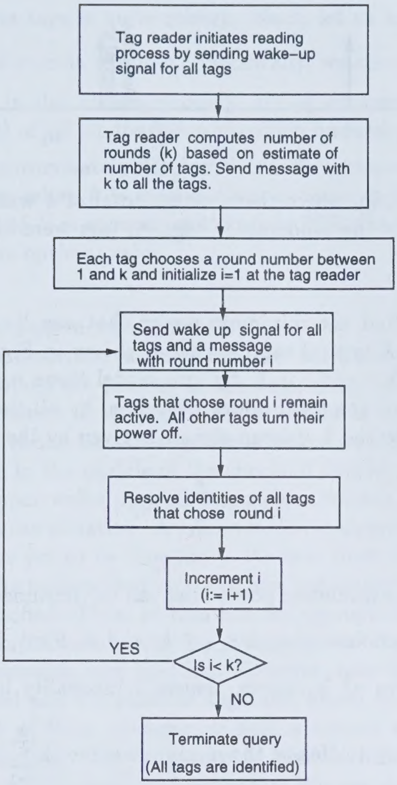

Tag reader initiates reading process by sending wake-up signal for all tags
Tag reader computes number of rounds (k) based on estimate of number of tags. Send message with k to all the tags.
Each tag chooses a round number between 1 and k and initialize i=1 at the tag reader
Send wake up signal for all tags and a message with round number i
Tags that chose round i remain active. All other tags turn their power off.
Resolve identities of all tags that chose round i
Increment i (i:= i+1)
YES
Is i< k?
NO
Terminate query (All tags are identified)

Figure 1: The basic algorithm to resolve the identities of N tags within the range of the tag reader.

choice of this value is motivated by the analysis of the following theorem, which will also show that our algorithm minimizes the power consumption per tag.

Observe that a given tag lies in a round with a total of $j-1$ other tags, where $\Pr(j-1) = \binom{N-1}{j-1}\left(\frac{1}{k}\right)^{j-1}\left(1-\frac{1}{k}\right)^{N-j}$. The power consumed by this tag in the round that it is identified is γj by the description given in the previous section. Hence, with our proposed algorithm, the power consumed per tag, on average, is

$$k\beta + \sum_{j=1}^{N} \binom{N-1}{j-1}\left(\frac{1}{k}\right)^{j-1}\left(1-\frac{1}{k}\right)^{N-j}\gamma j, \tag{2}$$

which can easily be shown to equal $k\beta + \gamma\frac{N-1}{k} + \gamma$. On substituting the value of k above, the average power per tag can seen to be $\sqrt{\gamma\beta N} + \sqrt{\gamma\beta N}\left(1-\frac{1}{N}\right) + \gamma$.

Theorem 1 *The minimum power per tag when there are N tags whose identities are to be resolved is $2\sqrt{\gamma\beta N}$.*

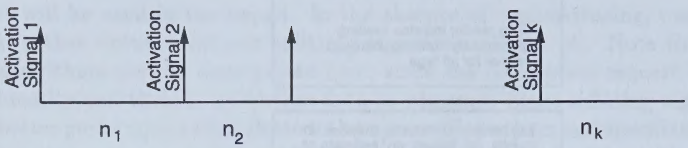


Figure 2: A typical scenario, where there were a total of k wakeup signals transmitted by the base station and of the unidentified tags, n_i tags were identified between signals i and $i+1$.

Proof: Let us find the minimum power that can be consumed for resolving the identities of n tags. A typical sample path is shown in Figure 2. Since n_i tags are identified between signals i and $i+1$, by our model these n_i tags each consume, on average, γn_i power. Thus, the total power consumed by all the tags for the resolution process and in response to the k wakeup signals is given by the following expression,

$$Nk\beta + \sum_{i=1}^{k} \gamma (n_i)^2, \tag{3}$$

subject to $\sum_{i=1}^{k} n_i = N$. The minimum power that can be consumed is found by minimizing Eq. (3) over all possible choices of $n_i, 1 \leq i \leq k$, and k. First, fixing k, let us minimize $\sum_{i=1}^{k} (n_i)^2$. Since, each term n_i^2 is convex, Jensen's Inequality implies that this term is minimized when $|n_i - n_j| \leq 1$. Hence, the minimum value of $\sum_{i=1}^{k} (n_i)^2$ is given by $k(N/k)^2$ (actually, the minimum value with the integer constraints taken into account will be a little higher, but we can ignore that for the moment). Thus, the minimum value of Equation (3) for a given value of k is given by $Nk\beta + \gamma \frac{N^2}{k}$. Minimizing this expression over k, yields that $k = \sqrt{\frac{\gamma N}{\beta}}$, and that the minimum average power consumption per tag, obtained by dividing the total power given in Equation (3) by N is $2\sqrt{\gamma \beta N}$. $\square$

Robustness with estimate of N:
The reader may notice that an explicit knowledge of the number of tags, N, is required for the algorithm. However, as long as N is known within a factor of 2 the performance of the algorithm is still close to optimal. The first observation is that the number of batches, m, chosen for the n active tags according to the procedure in Section 2 (as $m = 0.786n$) does not have to be precise. Analysis and numerical experiments confirm this fact. For example, for $n = 100$ a 50% error in the value of m (equivalently in the estimate of n) causes only a 7% increase in the power consumed per tag, i.e., the effective value of γ goes to 1.12. For $n = 10$ a 50% error in the value of m causes a 8% increase in the effective value of γ. We shall now assume that a 50% error in the value of m causes at most a 10% error in the effective value of γ. Thus, the procedure that we

use each time we activate tags is quite robust. Next, let us examine what happens if we choose the number of rounds wrongly. Specifically, we choose $k = \sqrt{\frac{\gamma 2N}{\beta}}$ rounds and then as a result m is also chosen wrongly, giving an average power consumption according to Equation (2) of $k\beta + 1.1(\gamma\frac{N-1}{k} + \gamma)$, which on further substitution leads to a power consumption of approximately $2.2\sqrt{\gamma\beta N}$, a 10% increase over the optimal value. A similar analysis applies when k is smaller than optimal as well. Thus, we conclude that even if the estimate of N is accurate within only 50%, the power consumption is at most 10% higher than the optimal value.

Online Versions (or iterative versions):
The algorithms we have described so far cover the *batch* case, that is all tags whose identities have to be resolved are present at the beginning. An interesting variation is one where some tags are known at the beginning, but some more are added in the middle of the procedure. A motivation for this scenario is where a shopper comes to a checkout (i.e., inventory) and then in the middle of the checkout decides to add some more items (or a partner of the shopper walks up with some more items!). Note that this scenario considerably complicates the situation. A classic iterative algorithm would only take into account the tags that are yet to be identified. We now have to not only minimize the power consumption for the unidentified tags, but also not cause too many wakeup signals for the tags already identified. Thus, in this case an appropriate cost function has first to be identified. For example, minimizing the average power consumed per tag from the new epoch may be a reasonable cost function. However, note that this quantity will be different for the identified and unidentified tags and hence an appropriate weight will have to be given to each of these to come up with a correct algorithm. Suppose that at some stage we have n_1 tags that have already been identified and n_2 tags that are yet to be identified in the vicinity of the tag reader. We can define functions $P_1(n_1, n_2)$ and $P_2(n_1, n_2)$ as being the average power consumption from this starting scenario of the n_1 identified and n_2 unidentified tags respectively. Then, a dynamic programming procedure can be used to determine the optimal number of rounds from now on. We will not go into the details here, but simply note that the above-described procedure is easy to implement.

4 Multiple Wakeup Signals

In this section, we explore the case when there are c possible signals[1] on which to make tags active, and that a given tag will be made active on only one of these c signals. Further, once a tag has been made active the tag reader can remotely set the next signal that the tag will respond to. In this environment it is possible to devise several algorithms to minimize the power consumption of tags. In all cases, when we refer to resolving the identities of tags using one signal only, we refer to the methods described in Section 3. We will briefly describe two algorithms below, whose performance is captured in Theorem 2.

Batched resolution: In this algorithm the N tags are split into c batches in the first round. Each batch of average size N/c tags is resolved separately using signals on one of

---

[1] This model can be realized quite easily by having a (small) set of antennae on the tag, each one tuned to a different frequency and the signals going out on these frequencies, respectively.

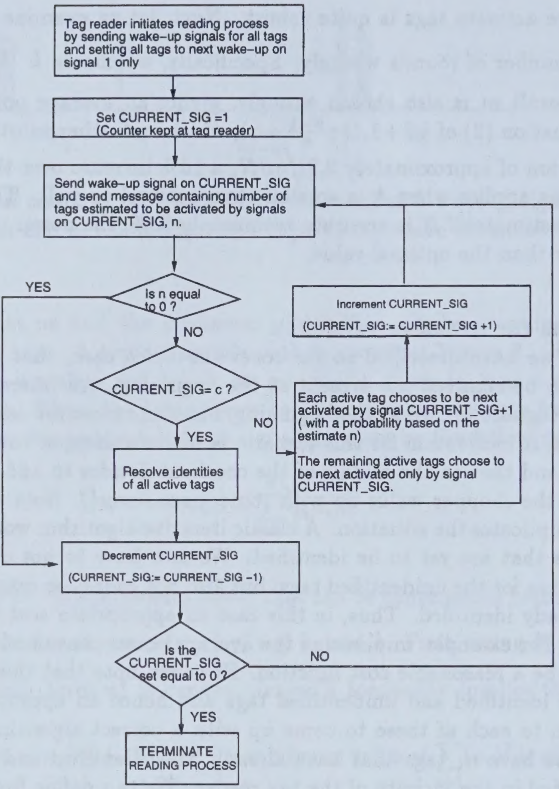

Tag reader initiates reading process by sending wake-up signals for all tags and setting all tags to next wake-up on signal 1 only
Set CURRENT_SIG =1 (Counter kept at tag reader)
Send wake-up signal on CURRENT_SIG and send message containing number of tags estimated to be activated by signals on CURRENT_SIG, n.
YES
Is n equal to 0 ?
NO
Increment CURRENT_SIG (CURRENT_SIG:= CURRENT_SIG +1)
CURRENT_SIG= c ?
NO
YES
Each active tag chooses to be next activated by signal CURRENT_SIG+1 (with a probability based on the estimate n)
The remaining active tags choose to be next activated only by signal CURRENT_SIG.
Resolve identities of all active tags
Decrement CURRENT_SIG (CURRENT_SIG:= CURRENT_SIG -1)
Is the CURRENT_SIG set equal to 0 ?
NO
YES
TERMINATE READING PROCESS

Figure 3: The basic algorithm to resolve the identities of N tags within the range of the tag reader with multiple signals.

the c signals only. The total power consumption per tag can then be easily determined to be $\beta + 2\sqrt{\gamma\beta\frac{N}{c}}$.

Tail resolution: This algorithm is best described via recursion (and proved by induction) and is shown in Figure 3. Enumerate the c signals as $1, 2, \ldots, c$. The basis case with $c = 1$ has been described before as the single frequency resolution procedure. Now, suppose that there are c signals. Wake up all tags and set a subset n_1 of them to respond to signals $2, 3, \ldots, c$ and the remainder to signal 1. Do the best possible power saving technique to resolve these n_1 tags using signals $2, 3 \ldots, c$ only. At the end of this resolution, make all tags active and then choose the next n_1 tags and proceed as before. Set the original set of n_1 tags to respond to signal 1 only.

Theorem 2 *The power consumption per tag when there are N tags whose identities are to be resolved and their are c distinct signals to activate tags is less than*

$$\min\left(\beta + 2\sqrt{\gamma\beta\frac{N}{c}},\quad (c+1)\gamma^{\frac{1}{1+c}}\beta^{1-\frac{1}{c+1}}N^{\frac{1}{c+1}}\right).$$

We outline a proof of this theorem, by construction. In brief, the two distinct algorithms each yield a power consumption per tag that corresponds to the terms in the min respectively.

We shall show that for the tail resolution procedure the best choice for n_1 is given by $n_1 = \gamma^{-\frac{1}{1+c}}\beta^{\frac{1}{c+1}}N^{\frac{c}{c+1}}$ and that with this choice of n_1, the theorem will hold. By our induction hypothesis, the net power consumption per tag with $c-1$ signals, when there are a total of N tags, is less than $c\gamma^{\frac{1}{c}}\beta^{1-\frac{1}{c}}N^{\frac{1}{c}}$. The basis case with $c = 1$ gives $2\sqrt{\gamma\beta N}$, which holds by Theorem 1 and the results of the previous section. Thus, each tag will choose with probability n_1/N to be active in a given round and there will be a total of N/n_1 rounds, i.e., instances when wakeup signals on signal 1 will be transmitted. Without showing all the details, it is intuitively clear that the average power consumed per tag will be $\frac{N}{n_1}\beta + c\gamma^{\frac{1}{c}}\beta^{1-\frac{1}{c}}(n_1)^{\frac{1}{c}}$, which can be shown to be minimized at the value of n_1 given above. Substitution of this value yields that the power consumption per tag is $(c+1)\gamma^{\frac{1}{1+c}}\beta^{1-\frac{1}{c+1}}N^{\frac{1}{c+1}}$, proving the remainder of the theorem.

Comparison: The relative efficiency of the batched resolution algorithm and the tail resolution algorithm depends upon the numerical values of c, β, γ and N. While for very large N and small β, γ the tail algorithm is better, the situation may be reversed for moderate values of N. Assuming $c = 4$, $\beta = 4$, $\gamma = 1$ and $N = 100$, the batched resolution algorithm requires $4 + 25 = 29$ units of power whereas the tail resolution algorithm requires $5 \cdot 100^{0.2} \cdot 4^{0.8} = 38.07$ units of power.

5 Conclusion

We have shown that taking advantage of the ability to remotely activate and de-activate the identification tags can significantly reduce their power consumption. Specifically, the results of the last two sections can be summarized to state that if N tags have to be identified and there are c distinct signals with which to activate tags, then the power consumption is proportional to $N^{\frac{1}{c+1}}$. Further, this reduced power consumption did not adversely impact the overall time to identify all the tags. In the case when there was only one signal ($c = 1$), we were able to prove that this was the minimum power consumption. However, for $c > 1$, we were not able to show that this was the best possible, and the methods of $c = 1$ do not extend in a straightforward fashion to this case. However, we do conjecture that the results stated in Theorem 2 are, in fact, optimal. Extending the results from a batch problem to one where more tags that have to be identified are brought in the middle of the procedure is also possible and methods to attack that problem were described. We believe that this work is a significant step in addressing the topic of saving power through contention resolution protocols, and that such methods can play an important role in many many emerging PCS systems.

References

[1] D. Bertsekas and R. Gallager, *Data Networks.* Englewood Cliffs, NJ: Prentice-Hall, second ed., 1992.

[2] R. Rom and M. Sidi, *Multiple Access Protocols: Performance and Analysis.* Springer Verlag, 1990.

[3] S. P. Curtis, "Transponder technologies, applications, and benefits," in *IEE Colloqium on 'The Use of Electronic Transponders in Automation ' (Digest No. 23)*, (London, England), pp. 2/1–8, IEE, February 15 1989.

[4] L. D. Bradley, E. N. Deagle, S. Glover, and S. Harris, "Radio frequency baggage tags," 1994. Hughes Microelectronics Europa Ltd., European Patent EP595549-A2.

[5] R. E. Floyd, "Radio frequency identification," in *Proc. of IEEE SOUTHEASTCON*, IEEE Press, 1993.

[6] D. J. Hind and D. Serby, "Radio frequency identification and tracking systems in hazardous areas," in *Proc. 5th International Conference on Electrical Safety in Hazardous Environments, IEE Conference Publication*, no. 390, pp. 215–227, IEE, Michael Faraday House, Stevenage, England, 1994.

[7] A. von Jena, B. Fischer, V. Magori, and W. Talke, "Intelligent sensor for monitoring freight waggon working conditions," *Sens. Actuators A, Phys. (Switzerland)*, vol. A42, no. 1-3, pp. 347–53, 15 April 1994.

[8] R. F. Cunningham, "Smart card applications in integrated transit fare, parking fee and automated toll payment systems the maps concept," in *Proceedings of the IEEE National Telesystems Conference*, pp. 21–25, IEEE Press, 1993.

[9] L. Reindl and W. Ruile, "Programmable reflectors for saw-id-tags," in *Proc. IEEE Ultrasonics Symposium*, pp. 125–130, IEEE Press, 1993.

[10] U. Kaiser and W. Steinhagen, "Low power transponder IC for high performance identification systems," in *Proc. Custom Integrated Circuits Conference*, pp. 335–338, IEEE Press, 1994.

[11] D. D. Kandlur and A. Krishna, "A power-efficient technique for multiple tag discrimination." U.S. Patent filed April 1995, Docket No. Y0995088 on behalf of IBM Corp.

[12] C. A. Warwick, "Trends and limits in the "talk time" of personal communicators," *Proceedings of the IEEE*, vol. 83, pp. 681–686, April 1995.

[13] A. A. Abidi, "Low-power radio frequency IC's for portable communications," *Proceedings of the IEEE*, vol. 83, pp. 544–569, April 1995.

[14] U. Madhow, M. L. Honig, and K. Steiglitz, "Optimization of wireless resources for personal communications mobility tracking," in *Proceedings of IEEE INFOCOM '94, The Conference on Computer Communications*, (Toronto, Canada), pp. 577–84, IEEE Computer Society Press, June 12-16 1994.

[15] M. Mouly and M. Pautet, *The GSM system for Mobile Communications.* 49, Rue Louise Bruneau, F-91120 Palaiseau, France: M. Mouly and M.-B. Pautet, 1992. ISBN 2-9507190-0-7.

[16] I. Cidon and M. Sidi, "Conflict multiplicity estimation and batch resolution algorithms," *IEEE Transaction Information Theory*, pp. 101–110, January 1988.

An Algorithm for Minimizing Queuing Delay of Packets in Cellular Data Networks*

John M. Harris and Srikanta P. Kumar
Northwestern University
2145 Sheridan Road
Evanston, Illinois 60208-3118
{jharris, kumars}@eecs.nwu.edu

Abstract:

This paper presents the Minimum Delay Algorithm (MDA) which schedules each packet in a cellular packet data network for transmission to a base station. The schedule produced by the MDA has both the minimum flow-time delay and the minimum makespan (evacuation) delay possible. The algorithm assumes that associated with each packet is a specific set of the base stations to which the packet may be transmitted directly. This is the set of bases which are in range. This set can be uniquely specified for each packet in the MDA. The algorithm uses the Maximum Cardinality Matching Algorithm (MCMA) in an iterative fashion. The MDA concludes in $O(m^3)$ steps where m is the number of packets to be scheduled.

1.0) Introduction:

Cellular packet data networks (such as CDPD, ARDIS data network) have several design issues that are different than those encountered in cellular voice networks. In cellular packet data networks, customers may experience delay. Customers are sensitive to this delay. Also, in cellular packet data networks, handoffs arising from cell crossings during transmission are less common. This paper presents an efficient algorithm for finding a minimum delay schedule for a given set of packets.

Coffman [1] (see also [2], [6], and [8]) considered optimal scheduling in multihop radio networks (primarily relevant in military applications) where a packet may go through multiple intermediate bases before it reaches it's destination. He showed that minimizing the makespan delay in such a multihop network, where the schedule must be computed at each interconnected base, is NP complete. However, the cellular data network problem is simpler in the sense that once a packet reaches a base station it will be delivered to its final destination via the wired network. The delay incurred in the wired network is typically negligible compared to the delay in the wireless network. Scarcity of bandwidth in cellular networks makes the wireless transmission of a packet significantly more costly than that in the wired network. For this reason, we seek to minimize the delay in the wireless portion of the packets transmission path.

2.0) Problem Description:

This problem can be represented by a bipartite graph G=(P,B,A). The bipartite graph contains the two disjoint sets of nodes P and B. P is the set of packets to be scheduled. B is the set of base stations. P(i) is the node representing packet i and B(j) is the node representing base station channel j. This algorithm assumes that associated with each packet is a specific set of the base station channels which are in range and to which the packet may be transmitted directly.

The Minimum Delay Algorithm (MDA) considered in this paper allows cells in the cellular packet data network to overlap. See figure 1. R meters is the radius of each cell in the network, because R is equal to the maximum transmission range of each packet sender. C_i is defined to be the set of base stations which are within R meters of packet i. Since packet i is within transmission range of the base stations in the set C_i, packet i can transmit to any

---

[*] This work was partially supported by both the Daniel E. Nobel IEEE Vehicular Technology Society/Motorola Fellowship and by the Armed Forces Communications and Electronics Association (AFCEA) Educational Foundation Fellowship.

channels assigned to station channel in the set C_i. All cells whose centers are closer than 4R meters are assumed to use distinct channels/frequencies.[1] This, and the fact that the Minimum Delay Algorithm (MDA) does not allow any two packets to transmit to the same base station channel at the same time guarantees that the concurrent transmission of packets within the same cell do not interfere. This is because, by definition, transmissions within the same cell will always be directed towards distinct overlapping cells, and thus will occur on distinct channels.

If base station j has access to k channels, then the base station j can be represented in the MDA by k different one channel base stations with the exact same service area. All packets within R meters of base station j may transmit to any of the k one channel base stations which base station j has access to. For ease of reference, in this paper, *base station channels will simply be referred to as base stations*. Each base station is assigned a unique number between 1 and n, where n is the number of base stations. Similarly, each packet is assigned a number between 1 and m where m is the number of packets.

In figure 2, the scheduling problem from figure 1 is represented by the graph G=(P,B,A). P(i) and B(j) are the packet and base nodes in the graph, respectively. The set of arcs A, between nodes in the set P and nodes in the set B, contains the information on which packets can transmitted to which base stations. If and only if the arc $(P(i),B(j)) \in A$, can packet i be transmitted to base station j. In other words, the arc $(P(i),B(j)) \in A$ if and only if base station j $\in C_i$. The MDA allows the set A to contain any combination of the (m*n) arc pairs possible. All arcs in this paper are undirected.[2]

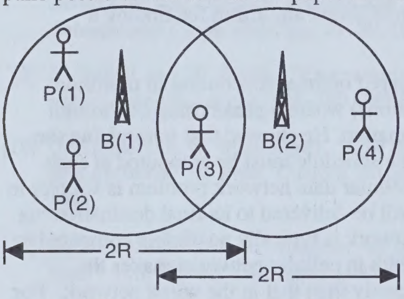


Figure 1:Overlapping Cells

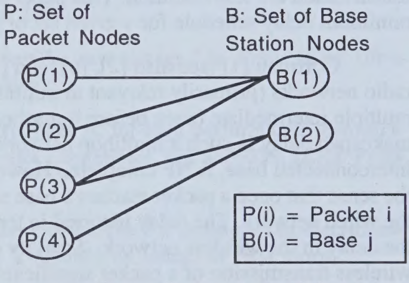


Figure 2: Graph G=(P,B,A)

In figure 2, the graph G=(P,B,A) corresponding to the problem presented in figure 1 is depicted. It contains the four packets in the set P={P(1),P(2),P(3),P(4)}, the two base stations in B={B(1),B(2)}, and the five arcs in the set A={(P(1),B(1)), (P(2),B(1)), (P(3),B(1)), (P(3),B(2)), (P(4),B(2))}. Packet 3 can be sent to either base station 1 or 2, packets 1 and 2 can be sent to base station 1 and packet 4 can be sent to base station 2. This is because each packet sender has a maximum transmission range of R meters and the packet senders are located as depicted in figure 1.

The MDA assumes that each packet requires one unit of time, or one *time slot*, for transmission. Additionally, it is assumed that all transmissions are synchronous. In the schedule, each packet must be matched with a base station and a time slot.

The goal of the MDA is to produce the minimum flow-time delay scheduling of the packets. The flow-time delay is the sum of the delays experienced by each of the individual packets. The minimum flow-time delay scheduling yielded by the MDA is also shown to be the minimum makespan delay (evacuation time) scheduling.

---

[1] i.e. TDM time slot or FDM frequency.

[2] The arc (i,j) is *incident* to both node i and node j.

2.1 Extended Graph:

Each packet needs to be scheduled for not just a base station, but also for a time slot at that base station. For this reason, the bipartite graph G=(P,B,A) needs to be extended to include separate nodes for each time slot at each base station. The resulting bipartite graph is $G_t=(P,B_t,A_t)$. The node B(j,k), in the node set B_t, denotes the reservation for base station j during time slot k. The set B_t contains the n*t nodes, for the n base stations during each of the first t time slots. A_t is defined in terms of the arc set A. The arc $(P(i),B(j,k))\in A_t$ if and only if the arc $(P(i),B(j)) \in A$, and k is an integer such that $0\le k\le t$. In other words, the arc $(P(i),B(j,k))\in A_t$ if packet i can transmit to base j and $0\le k\le t$. The graph G=(P,B,A), from figure 2, is generalized to represent individual time slots in the graph $G_2=(P,B_2,A_2)$ in Figure 3.

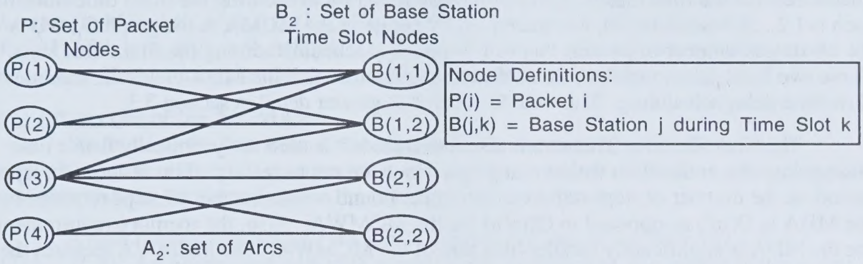


Figure 3: $G_2=(P,B_2,A_2)$

Note that, at most, m time slots are required to schedule m packets. The packet scheduling problem now reduces to finding a set $M \subseteq A_m$, such that the set M has certain properties. Here, M represents the schedule for the m packets during the first m time slots. If the arc $(P(i),B(j,k))\in M$, then the packet i is scheduled for transmission to the base station j during the time slot k.

The set M must be a matching. A *matching* is a set of arcs in which no two arcs are incident to the same node. M must be a matching for two reasons. First, each packet only needs to be transmitted once, therefore no two arcs which are incident to the same packet node can be in the set M. Also, because no two packets may be transmitted to the same base station during the same time slot, no two arcs in the set M can be incident to the same base station time slot node. If the arc $(P(i),B(j,k))\in M$, then P(j) and B(j,k) are said to be *matched.*

Any set $M \subseteq A_m$, where M is a matching defines a valid *schedule*. The set M_{MIN} is the minimum delay scheduling. M_{MIN} must contain m arcs, because all m of the packets must be scheduled. Also, M_{MIN} should have the minimum flow-time delay of all possible matchings M of m arcs, $M \subseteq A_m$. The flow-time delay of the set M_{MIN} is equal to the sum of the delays incurred by each of the packets in the schedule. In other words:

$$\text{Flow Time Delay}(M_{MIN}) = \sum_{(P(i),B(j,k))\in M_{MIN}} (k)$$

2.2) Discussion of Alternative Approaches:

The minimum delay problem could be solved directly by including the arc weight k for each arc (P(i),B(j,k)) in the graph $G_m=(P,B_m,A_m)$ and then applying the Maximum Cardinality Minimum Weight Matching Algorithm (MCMWA) to the graph G_m. See [4]. The MCMWA produces the maximum cardinality matching with the minimum weight possible. Here, the maximum cardinality matching of the graph G_m is m, the number of packets. Since the weight of each arc is equal to the delay incurred by including it in the matching, the weight of a matching is equal to it's flow-time delay. Thus, the maximum cardinality minimum weight matching is the minimum flow-time delay matching of m arcs.

The MDA, we present in this paper, exploits redundancies in the MCMWA. As a result, the MDA uses many fewer steps and much less memory. The MDA does not use arc weights. Rather, it uses the Maximum Cardinality Matching Algorithm (MCMA) [3] in an iterative fashion. The MCMA finds the maximum cardinality matching for the graph it is given. The MDA applies the MCMA to G_1 and produces the matching M_1. This matching is then used as the initial matching when the MCMA is applied to G_2. During the i^{th} cycle of the MDA, the matching M_{i-1} from the previous cycle is used as the initial matching by MCMA for the graph G_i. The MDA stops during the T^{th} of the MCMA when all of the packets have been scheduled.

2.3) MDA Approach Overview

Arc weights are not necessary in the MDA because of the Node Retention Theorem. It guarantees that the final matching M_T will transmit $|M_t|$ packets during the first t time slots, for each t=1,2,..,T.[3] Since the M_t was generated by applying the MCMA to the graph G_t, $|M_t|$ is the maximum number of packets that can be possibly scheduled during the first t time slots. These two facts, taken together, are sufficient to guarantee that the MDA yields the minimum flow-time delay scheduling. This will be covered in greater detail in section 3.3.

The Node Deletion Theorem is also introduced. It is used to dynamically delete base station time slot nodes from the current graph which are not necessary. This reduces the upper bound on the number of steps required. An upper bound on the number of steps required by the MDA is $O(m^3)$ as opposed to $O(m^3n)$ for the MCMWA. Also, the memory requirement for the MDA is significantly smaller than that of the MCMWA. The MCMWA requires the use of $O(m^2n)$ arc weights and the upkeep of more than 2mn variables. The MDA has fewer than m local variables, and does not use arc weights.

3.0) Description of the MCMA:

Some terminology is introduced next. Node k is *covered* by the matching M_t if there is an arc in the matching M_t which has node k as one of it's end points. If node k is covered by M_t, then node k is *matched* with the node $M_t(k)$ and the arc $(k, M_t(k)) \in M_t$.[4] An unmatched node is said to be *exposed.* The *domain of the matching M_t* is the set of packet nodes covered by M_t. The *range of the matching M_t* is the sets of base station time slot nodes covered by M_t.

The MCMA, was developed by Ford and Fulkerson [3]. The version of the algorithm below is from Rockafeller [7]. See also, [4] and [5]. When the MCMA is applied to the graph $G_t=(P,B_t,A_t)$, beginning with any matching M_{t-1}, it yields A the largest matching M_t possible in the graph G_t. In other words, the algorithm yields a scheduling M_t, which schedules the maximum number of packets possible, during the first t time slots. Note that W\X denotes set subtraction of X from W. In other words, $x \in (W \backslash X)$ if and only if $x \in W$ AND $x \notin X$.

```
Maximum Cardinality Matching Algorithm (MCMA): Ford [3], Rockafeller [7]
Given:   Graph $G_t = (P,B_t,A_t)$.
         $M_{t-1}$ = Initial matching, which may be the empty set.
         Initialize $M_t = M_{t-1}$.
Step 0:  Initialization: The variables E and W, and the function $\varphi()$ are local to the MCMA.
           Set E = Set of packets covered by $M_t$ = domain $M_t$
           Set W = Set of packets not covered by $M_t$ = $P \backslash (\text{domain } M_t)$
           The function $\varphi()$ is undefined.
Step 1:  Exit Condition:
         Is W empty?.
           If yes, then $M_t$ solves the maximum matching problem. Stop.
           If no, then go to step 2.
```

---

[3] |W| denotes the cardinality of (number of elements in) the set W.

[4] In a slight abuse of notation, when M_t appears without an argument, it refers to the set of arcs in the matching. $M_t(k)$, however, refers to the base station time slot node which is matched with packet i.

Step 2: *Select a Node to Scan:*
Select any $k \in W$. *Does there exist a* $l \in B_t \setminus (\text{range } M_t)$ *with* $(k,l) \in A_t$?
If yes, then go to step 4.
If no, then go to step 3.

Step 3 *Label Nodes:*
For each $k' \in E$ *with the arc* $(k,M_t(k')) \in A_t$, *transfer* k' *from E to W, and define* $\varphi(k') = k$. *Delete k from W and return to step 1.*

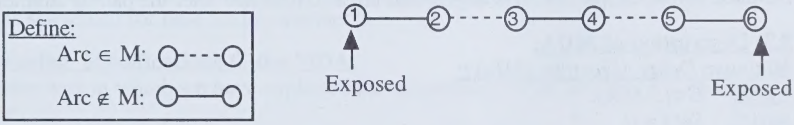
Step 4: *Augment the Augmenting Path:*
Generate the sequence $k_0,k_1,\ldots,k_r$, *where* $k_0=k$, $k_i=\varphi(k_{i-1})$ *for* $i=1,\ldots,r$, *and* k_r *is not in the domain of* φ. *If k itself is not in the domain of* φ, *the sequence consists of just* $k = k_0$. *Delete the arcs* $(k_0,M_t(k_0)),(k_1,M_t(k_1)),\ldots,(k_{r-1},M_t(k_{r-1})),(k_r,M_t(k_r))$ *from* M_t *and add the arcs* $(k_0,l),(k_1,M_t(k_0)),(k_2,M_t(k_1)),(k_3,M_t(k_2))\ldots,(k_{r-1},M_t(k_{r-2})),(k_r,M_t(k_{r-1}))$ *to* M_t. *The resulting set* M_t *is also a matching with cardinality increased by one. Return to step 0.*

3.1) Properties of the MCMA:

The basis of the operation of the MCMA is the augmenting path. When an augmenting path is found, it is augmented, as was done in step 4. When an augmenting path is augmented, the number of arcs in the matching increases by one.

A *path* between the nodes 1 and 6 is a sequence of arcs of the form $(1,i_1)$, (i_1,i_2), (i_2,i_3),...,$(i_k,6)$. An *alternating path* with respect to the matching M_t is a path in which every other arc in the path is in the set M_t. An *augmenting path* is an alternating path between two exposed (unmatched) nodes. An example of an augmenting path is in figure 4. As seen in figure 4, the number of arcs in the matching M_t is increased by one when the augmenting path is augmented. When an augmenting path is augmented, all of the arcs in the path that were originally in M_t, are removed from M_t, and all of the arcs in the path which were not originally in M_t are added to M_t. Since, the arcs on the two ends of augmenting path are not matched, an augmented path always has cardinality greater by one.

Before the augmenting path (1,6) is augmented, **M contains 2 arcs:**



After the augmenting path (1,6) is augmented, **M contains 3 arcs:**

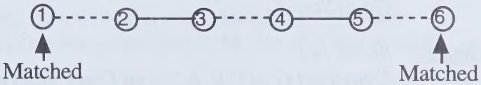


Figure 4: Augmenting Path is Augmented

Augmenting Path Theorem: [3]

A matching M_t contains a maximum number of arcs if and only if it does not contain any augmenting paths.

Proof: For a formal proof, see page 13 of [3]. ■

This theorem holds since, as in Figure 4, the presence of an augmenting path means that the number of arcs in the set M_t can be increased by one. The MCMA performs an efficient, exhaustive search for an augmenting path in steps one and two. When an augmenting path is found, it is augmented in step 4 of the MCMA. If no augmenting path is immediately accessible, nodes are labeled in step 3, so that the existing augmenting paths can be extended. This process is repeated until no more augmenting paths exist, and thus the matching has maximum cardinality.

Here, we extract properties inherent in the process of augmenting a path. First, is the MCMA Node Retention Theorem. It states:

MCMA Node Retention Theorem:

Every node covered by the initial matching, M_{t-1}, of the MCMA will also be covered in the maximum cardinality matching M_t which the MCMA yields.

In other words:

$$\text{Domain } (M_{t-1}) \subseteq \text{Domain } (M_t) \qquad \text{(eqn 1)}$$

$$\text{Range } (M_{t-1}) \subseteq \text{Range } (M_t) \qquad \text{(eqn 2)}$$

Note that: $N_t = |\text{Domain } (M_t)| = |\text{Range } (M_t)| = |M_t|$ = maximum number of packets that can possibly be scheduled during the first t time slots.

Proof:

The only time arcs are removed from or added to M_t in the MCMA is in step 4 when the path $k_0, M_t(k_0), k_1, M_t(k_1), \ldots, k_{r-1}, M_t(k_{r-1})$ is augmented. During step 4 of the MCMA, the operations performed are:

Nodes Originally Matched: $k_0, M_t(k_0), k_1, M_t(k_1), \ldots, k_{r-1}, M_t(k_{r-1})$

Arcs Deleted: $(k_0,M_t(k_0)), (k_1,M_t(k_1)) \ldots, (k_{r-1},M_t(k_{r-1}))$

Arcs Added: $(k_0,l), (k_1,M_t(k_0)), (k_2,M_t(k_1)),\ldots, (k_r,M_t(k_{r-1}))$

Nodes Matched after Augmentation: $l, k_0, M_t(k_0), k_1, M_t(k_1),\ldots, k_{r-1}, M_t(k_{r-1})$

It is clear that all of the nodes which were originally matched, $k_0, M_t(k_0), k_1, M_t(k_1), \ldots, k_{r-1}, M_t(k_{r-1})$, are still matched after the augmenting path is augmented. This is because each of these nodes appear as the end point in one of the arcs which is added to the matching M_t.

It is also apparent that node retention holds in figure 4. Nodes 2, 3, 4, and 5 which are matched before the path (1,6) is augmented are also matched after the path is augmented. ■

3.3) Description of MDA:

Minimum Delay Algorithm (MDA):

```
Given:  G=(P,B,A).
Step 0: Set t = 0.
        Set $M_0$ = {} = empty set
        Go to Step 1.

Step 1: Set t= t+1.
        Construct $G_t=(P,B_t,A_t)$ from G=(P,B,A) as described in section 2.1.
        Go to step 2.

Step 2: Run the MCMA on the graph $G_t=(P,B_t,A_t)$, starting with the initial set $M_{t-1}$. Place the
            resulting maximum cardinality set in $M_t$.
        Go to step 3.

Step 3: Does the number of arcs in $M_t$ equal the number of packets?
            If no, then go to step 1.
            If yes, then $M_t$ is the minimum flow-time and makespan delay
                scheduling/matching. STOP.
```

The MDA terminates at the first time slot t=T, such that all of the packets are scheduled. Since, the MDA functions in this way, T is the minimum number of time slots which can accommodate all of the packets. Thus, the MDA schedule M_T has the minimum makespan delay (evacuation time).

The MDA involves the iterative use of the MCMA. Each cycle of the MDA, starts with the initial matching M_{t-1} and yields the matching M_t. For this reason, the MCMA Node Retention Theorem can easily be generalized in to the MDA Node Retention Theorem.

MDA Node Retention Theorem:

In the MDA, if base or packet node x is matched in the set M_t, then the base or packet node x is also matched in the scheduling $M_{t'}$, for all t and t' such that $t \le t' \le T$. In other words, for all t and t' such that $t \le t' \le T$:

$$\text{Domain of } M_t \subseteq \text{Domain of } M_{t'} \text{ and} \qquad \text{(eqn 3)}$$

$$\text{Range of } M_t \subseteq \text{Range of } M_{t'} \qquad \text{(eqn 4)}$$

Proof:

If node x is in the set M_t, then node x is in set $M_{t'}$ for all t and t' such that: $t \le t' \le T$. This fact follows directly from the MCMA Node Retention Theorem and the iterative use of MCMA by the MDA. ■

Proof of the Minimum Delay Property of the MDA:

From the MDA Node Retention Theorem, we know that all of the nodes which are matched in the set M_t are also matched in final scheduling M_T for t=1,2,..,T. Thus,
1) the scheduling M_T which results from the MDA will have $|M_t|$ packets scheduled during the first t time slots for t=1,2,..,T. Also,
2) $|M_t|$ is guaranteed to be the maximum number of packets that can possibly be scheduled during the first t time slots by the MCMA and the Augmenting Path Theorem.

These facts are sufficient to guarantee that the scheduling M_T has the minimum flow-time delay and the minimum makespan delay possible. ■

In figure 5, in the appendix at the end of this paper, the MDA is applied to the problem described in Figures 1, 2 and 3. Note that all four packets can be scheduled in two time slots only if P(2) is scheduled for base station number 2.

4.0) Time Saving Modifications to the MDA:

Another way in which we have exploited the redundancy in the MCMWA is through the use of the :

Node Deletion Lemma:

If $B(j,t) \notin$ range of M_t then $B(j,t) \notin$ range in $M_{t'}$ for all t and t' such that $t \le t' \le T$. In other words, if B(j,t) is not scheduled in M_t then B(j,t) is not scheduled in $M_{t'}$ for all t and t' such that $t \le t' \le T$.[5]

Proof:

1) $|M_t|$ is the maximum number of base station time slot nodes that can be scheduled during the first t time slots.
2) From the Node Retention Theorem, it is known that the base station time slot nodes which are in the range of M_t are also in the range of $M_{t'}$.

There fore, the $|M_t|$ base station time slot nodes which are scheduled during the first t time slots in $M_{t'}$ are the same base station time slot nodes in the set M_t. ■

---

[5] By definition, if and only if B(j,t) is in the range of M_t, is B(j,t) scheduled in the matching M_t.

Node Deletion Theorem:

If B(j,t) is not matched in the scheduling set M_t, then B(j,t') node is not matched in the scheduling set $M_{t'}$, for all t and t' such that t≤t'≤T. In other words, if B(j,t) ∉ range of M_t, then B(j,t') ∉ range of $M_{t'}$ for all t and t' such that t≤t'≤T.

Proof By Contradiction:

Assertion: B(j,t) ∉ range of M_t and B(j,t') ∈ range of $M_{t'}$.

From the Node Deletion Lemma, this assertion is equivalent to:

B(j,t) ∉ range of $M_{t'}$ and B(j,t') ∈ range of $M_{t'}$.

P(x) is the packet which is matched with B(j,t'), B(j,t') ∈ range of $M_{t'}$. P(x) could be matched with B(j,t) instead of B(j,t') in the matching $M_{t'}$. This would increase: $|\text{range}(M_{t'}) \cap B_t|$ = number of matched base station time slots in the first t time slots of $M_{t'}$. P(x) can be matched with B(j,t) instead of B(j,t') because every packet which is connected to B(j,t) is also connected to B(j,t') for all t and t' and B(j,t) is known to be unmatched.

This implies that the range of $M_{t'}$ does not contain the maximum number of base station time slots possible during the first t time slots.

This is a contradiction. By definition, $|M_t|$ equals the maximum number of base time slot nodes that can be matched in the first t time slots. It is not possible for the assertion to be true and thus the Node Deletion Lemma is true. Proof by contradiction. ■

From the Node Deletion Theorem, if a base station node for time slot t is not matched during cycle t of the MDA, then the base station j nodes for later time slots will also not be matched. The MDA can be modified to exploit this. One can remove all of the base station nodes which are not matched in M_t from the graphs $G_{t'}$, for all t and t' such that t ≤ t' ≤T. The resulting algorithm still provides the minimum flow-time delay scheduling, but requires fewer steps.

This can be achieved by making the following two changes to the MDA presented in section 3.3.

In Step 0 of the MDA, *initialize U_B and U_A each with an empty set.*

Replace step 2 in the MDA with:

Step 2: *Run the MCMA on the graph $G_t=(P,(B_t \backslash U_B),(A_t \backslash U_A))$ instead of $G_t=(P,B_t,A_t)$, and start with the initial set M_{t-1}. Place the resulting maximum cardinality set in M_t.*
For all $u \in (B_t \backslash B_{t-1})$ and $u \notin \text{range}(M_t)$ place u in the set U_B.
Place all arcs $(i,j) \in A_t$ for which $j \in U_B$, in the set U_A.
Go to step 3.

From the Node Deletion Theorem it can be shown that the number of packets scheduled for each time slot is monotonically decreasing as t increases. In other words, ($|M_t|$ - $|M_{t-1}|$) is monotonically decreasing with t. This is because base stations which were not matched in the previous time slot, can not be matched in the current time slot.

There exists another redundancy in the MDA which can be exploited. The matching can not have more elements than there are base station time slot nodes in the graph. When $G_t=(P,(B_t \backslash U_B),(A_t \backslash U_A))$ in the MCMA used by the MDA. When $|M_t| = |B_t \backslash U_B|$, M_t has the maximum cardinality achievable for the graph $(P,(B_t \backslash U_B),(A_t \backslash U_A))$. Thus, this exit condition can be added to the MDA. Replace step 1 in the MCMA used by the MDA with:

Step 1: *Is W empty **or is** $|M_t| = |B_t \backslash U_B|$?*
If yes, then M_t solves the maximum cardinality matching problem. Stop.
If no, then go to step 2.

5.0) Upper Bound for Number of Steps in MDA:

An upper bound on the number of steps the MDA will perform in the process of finding the optimal scheduling for m packets on n base stations can be established. Define:

A_i = number of times an augmenting path is augmented during the ith cycle of the MDA.

Each time an augmenting path is augmented, the cardinality of the matching increases by one. For this reason, when the total number of times an augmenting path has been augmented equals m, the cardinality of the matching will equal m. When the cardinality of the matching equals m, the MDA stops. Thus, $\sum_{i=1}^{m} A_i = m$. Also, by inspection, $0 \le A_i \le m$ for all m.

At most, m packet nodes must be scanned in order to find an augmenting path. During the jth MDA cycle, each packet node has, at most, $\left(\sum_{i=1}^{j-1} A_i\right) + A_{j-1}$ arcs connected to it. This is because, the number of arcs is less than or equal to the number of base station time slot nodes in the current graph. From the Node Deletion Theorem, there are, at most, $\left(\sum_{i=1}^{j-1} A_i\right) + A_{j-1}$ base station time slot nodes in the graph $(P, B_t \backslash U_B, A_t \backslash U_A)$ which is used during the jth cycle of the MDA.

Since, $\sum_{i=1}^{m} A_i = m$ and $0 \le A_i \le m$ for all m:

$$\begin{pmatrix} \text{Number of Arcs Inspected per Node} \\ \text{Scanned During the j}^{\text{th}} \text{ MDA Cycle} \end{pmatrix} = \left[\left(\sum_{i=1}^{j-1} A_i\right) + A_{j-1}\right] \le 2m .$$

Thus, an upper bound on the total number of steps required to perform the MDA on m packets and n bases is:

$$\left[\sum_{i=1}^{m} A_i * m * (2m)\right] \le 2m^3$$

The process of finding m augmenting paths involves scanning m nodes and each node scanned has fewer than 2m arcs.

Therefore, the MDA is $O(m^3)$. In practice, the actual number of steps required may be much lower. However, it is difficult to capture all of the step saving aspects of the MDA in an upper bound.

6.0) Conclusions:

The minimum delay algorithm (MDA) provides a minimum delay scheduling for packets in a cellular packet data network. The optimal schedule for m packets can be found in $O(m^3)$ steps. This schedule has both the minimum flow-time (sum) delay and the minimum makespan (evacuation) delay possible. The algorithm is efficient and reasonably simple to implement. The radius of R of the cells can be arbitrarily set. Also, no constraint is placed on the location of the cells and the packets in the MDA.

We are currently finalizing work incorporating random packet arrivals and random packet service times and a form of hybrid channel allocation into the MDA.

7.0) References:

[1] E. G. Coffman, Jr., M.R. Garey, D. S. Johnson and A. S. Lapaugh, “Scheduling File Transfers,” SIAM J. Computing, (1985) Vol. 14, pp 744-780.

[2] B. Hajek, “Link Scheduling in Polynomial Time.” SIAM J. Computing, (1988) Vol. 34, No. 5, pg. 910.

[3] L. R. Ford and D. R. Fulkerson, “Flows in Networks,” Princeton University Press, 1962, pp 55-56.

[4] E. Lawler, "Combinatorial Optimization." Holt Rinehart and Winston, New York, 1976.
[5] E. Minieka, "Optimization Algorithms for Networks and Graphs," Marcel Dekker, Inc., New York, 1978.
[6] S. Papavassiliou and L. Tassiulas, "Joint Optimal Channel Base Station and Power Assignment," 28th annual Conference of Information Science and Systems, March 1994, Princeton University.
[7] R. Rockafeller, "Network Flows and Monotropic Optimization." John Wiley & Sons, New York, 1984.
[8] S. Xu, P. Mirchandani, S. Kumar, and R. Weber, "Dispatching Stochastic Jobs in Parallel Servers," Journal of Applied Probability, 1990.

8.0) Appendix:

Define: Arc $\in$ M: ○- - -○
Arc $\notin$ M: ○—○

MDA Cycle 1: MCMA on $G_1 = (P, B_1, A_1)$, $M_1 = \{\}$
Step 0: $E = \{\}$, $W = \{P(1), P(2), P(3), P(4)\}$,
$M_1 = \{\}$, φ undefined.
Step 1: W is non-empty and $|M_1| < |B_1|$.
Step 2: Choose $k = P(3) \in W$, $l = B(1,1)$
Step 4: Augment: $(P(3), B(1,1)) \in M_1$.

Step 0: $E = \{P(3)\}$, $W = \{P(1), P(2), P(4)\}$,
$M_1 = \{P(3), B(1,1))\}$, j undefined.
Step 1: W is non-empty and $|M_1| < |B_1|$.
Step 2: Choose $k = P(4) \in W$, $l = B(2,1)$
Step 4: Augment: $(P(4), B(2,1)) \in M_1$.

Step 0: $E = \{P(3), P(4)\}$, $W = \{P(1), P(2)\}$,
$M_1 = \{P(3), B(1,1)), (P(4), B(2,1)\}$,
φ undefined.
Step 1: $|M_1| = |B_1| = 2$. Stop MDA cycle 1.

MDA Cycle 2: MCMA on $G_2 = (P, B_2, A_2)$. $M_2 = M_1$
Step 0: $E = \{P(3), P(4)\}$, $W = \{P(1), P(2)\}$,
$M_2 = \{P(3), B(1,1)), (P(4), B(2,1)\}$,
φ undefined.
Step 1: W is non-empty and $|M_2| < |B_2|$.
Step 2: Choose $k = P(1) \in W$, $l = B(1,2)$.
Step 4: Augment: $(P(1), B(1,2)) \in M_2$.

Step 0: $E = \{P(1), P(3), P(4)\}$, $W = \{P(2)\}$,
$M_2 = \{P(3), B(1,1)), (P(4), B(2,1), (P(1), B(1,2))\}$, φ undefined.
Step 1: W is non-empty and $|M_2| < |B_2|$.
Step 2: Choose $k = P(2) \in W$, no l exists.
for $(k,l) \in A_2$. Goto Step 3.
Step 3: For each $k' \in E$, $(k, M_2(k')) \in M_2$, $\varphi(k') = k$ and transfer k' from E to W. Delete k from W.
For $k' = P(1)$, $\varphi(P(1)) = P(2)$.
For $k' = P(3)$, $\varphi(P(3)) = P(2)$.
$W = \{P(1), P(3)\}$, $B = \{P(4)\}$

Step 1: W is non-empty and $|M_2| < |B_2|$.
Step 2: Choose $k = k_0 = P(3) \in W$, $l = B(2,2)$.
Step 4: $\varphi(k_1) = \varphi(P(3)) = k_0 = P(2)$
Augment the path $k_0 = P(3)$, $k_1 = P(2)$. (r=1)
Delete Arc $(k_0, M(k_0)) = (P(3), B(1,1))$ from M_2.
Add Arcs $(k_0, l) = (P(3), B(2,2))$, and
$(k_1, M(k_0)) = (P(2), B(1,1))$ to M_2.
$M_2 = \{P(3), B(1,1)), (P(4), B(2,1), (P(1), B(1,2))\}$

Step 1: $|M_2| = 4 = |B_2|$. Stop MDA cycle 2.

MDA Cycle 3: $|M_2| = |P|$. Stop.

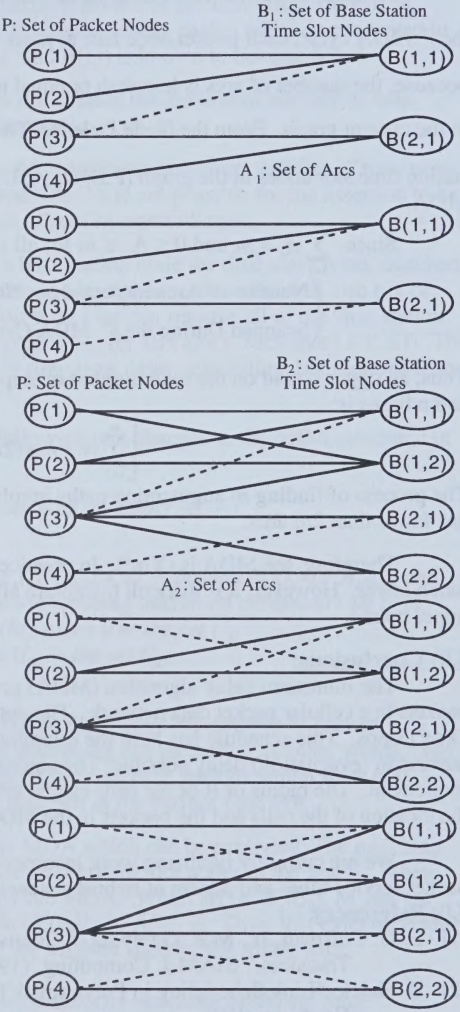


Figure 5: MDA Applied to Scheduling Problem in Figures 1, 2 and 3

Resource Allocation for Mobile Users in ATM Networks

Baiju V. Patel
IBM T. J. Watson Research Center
30 Saw Mill River Road
Hawthorne, NY 10532
baiju@watson.ibm.com

Mischa Schwartz*
Center for Telecommunication Research
Columbia University
New York, NY, 10027
schwartz@ctr.columbia.edu

October 4, 1995

Abstract

In this paper, we have proposed a tree based Virtual Embedded Network architecture to address resource reallocation problem for mobile users connected to an arbitrary topology ATM network. The approach helps us significantly reduce the resource reallocation rate on different links of ATM network. The architecture and results are presented for both linear and two dimensional topology of cells.

1 Introduction

Wireless and broadband ATM networks are considered to be the key technologies for communication in twenty-first century. The wireless networks will provide mobile users access to the rest of the world via wired ATM intrastructure. Such universal network will support traditional voice and data as well as emerging multimedia traffic [1, 2]. Quality of Service (QoS) provisioning plays a key role in supporting multimedia traffic. The combination of broadband wide area networks and wireless networks, with multimedia traffic transmitted across them, must reflect itself in QoS guarantees that combine aspects of QoS from both.

A user desiring to establish a multimedia session specifies the behavior using traffic descriptors, and specifies the desired quality of service. Based on this information, the network allocates resources in order to meet the QoS requirements. Much work is underway in defining the QoS parameters, traffic parameters, call admission and resource allocation for wired ATM networks. The QoS parameters include average cell delay, cell delay jitter and cell loss probability. The traffic descriptors include average cell transmission rate, peak cell transmission rate and number of cells that may be transmitted at peak rate (burst length)[3]. Based on these parameters, call admission and resource allocation algorithms are being developed essentially for switch-based ATM networks.

To sustain a multimedia session, the QoS must be guaranteed through the life of the session. This requires that resources be allocated in both the wired and wireless segments of the end-to-end connection. The resource allocation problem for the wireless portion of the end-to-end connection has been addressed in [4, 5, 6, 7] and in many other publications. Similarly, algorithms to allocate resources in ATM networks for stationary users have also been developed [8]. If each wireless user is treated identically to a wired user, the handoffs in moving from cell to cell will cause frequent setup and freeing of connections and associated resources. Therefore, the mobile user will have significant connection management overhead associated with it. When a user moves from cell to cell, the connection path in the wired network has to be modified to reflect the new location of the user. Some work has been done in this area[9, 10, 11, 12]. Since the mobile user poses a time variant environment to the ATM network, the static nature of resource allocation algorithms for ATM networks cannot be directly used for allocating resources to mobile users. This change in the path through the wired (e.g., ATM) network requires that new resources must be allocated on some links and resources be freed up on some other links. Thus resource management for the ATM network due to handoffs may potentially result in significant overhead once personal communication services (PCS) networks become prevalent. In this paper, we address the resource allocation problem arising in wired networks due to mobility of users in the wireless networks.

---

*This work was done while the author was visiting IBM T. J. Watson Research Center

The outline of the rest of the paper is as follows. In the next section, we define the tree-Virtual Embedded Network architecture. In Section 3, we study the handoff rate as seen by any switch of the tree-VEN for both linear and two-dimensional arrangements of access points. The final conclusions are presented in Section 4.

2 Architecture

With the emergence of ATM as the network of choice for both local and wide area communications, ATM networks are likely to be the choice for the wired network to facilitate the mobile users' communications with other mobile users or wired users[1, 2]. The concepts presented in this paper assume ATM wired networks, however, they do not make any specific assumptions about ATM network topology.

In the past, a hierarchical topology has been suggested for personal communication networks in [13]. Without loss of generality, we consider a three-level hierarchy as shown in Fig. 1. At the lowest level (the last mile) is the wireless cell. The wireless cells within a geographical area (e.g., city, county, highway) are connected by a regional network (middle level), and these regional networks are connected to the wide area network backbone (highest level). In practice, both the regional and backbone network may be implemented using ATM technology. It is assumed that regions are defined based on the population centers such that most of the mobile user activities are confined within a region (called neighboring mobile access region in [1]). For example, a regional network may cover an interstate highway, a suburb, or a town center. Most of the handoffs are assumed to take place within a region. Note that similar topologies exists for wireless local area networks where the regional network would be the campus or building network. Therefore, as the first step we will address the mobile users within a regional network, which will carry a significant fraction of the mobile traffic.

2.1 Virtual Embedded Network

Even though it is possible to embed VEN over an arbitrary-topology ATM network using a variety of topologies, in this paper we consider a tree topology for VEN (called tree-VEN) because it offers many attractive features. The tree topology has been widely used in solving routing problems in communication networks. For example, spanning trees are used to determine packet paths in bridged LANS[14]. Similarly, trees are used for determining routes in large IP networks (e.g., Internet)[15]. The tree topology is used to efficiently handle handoff routing in [1] where it is shown that the call processing overhead for trees embedded over an arbitrary regional network can significantly reduce the connection setup overhead due to handoff. We thus propose to use the tree topology for VEN over an arbitrary-topology regional ATM network. Since the algorithms to determine spanning/routing trees are well known, the establishment of

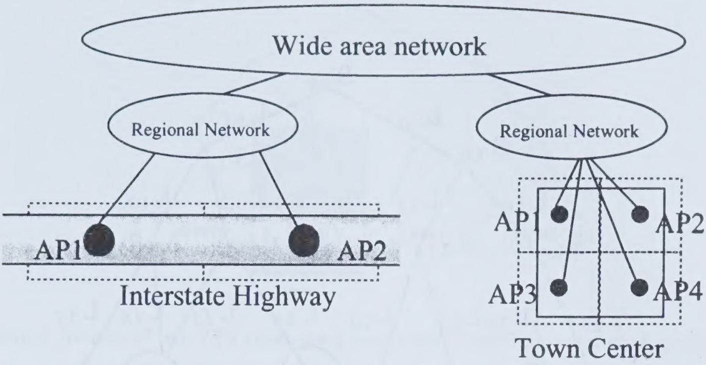


Figure 1: Hierarchical network under consideration.

the tree topology is not considered in this paper. We assume instead that a tree topology for a VEN has already been determined. Based on a given tree-VEN, we study the effect of handoffs on resource allocation on different links (levels) of a given tree-VEN. We further propose algorithms that reduce the resource reallocation overhead for tree-VEN. The actual information on the tree-VEN may be carried using an ATM connection hierarchy based on virtual paths (VP's) and virtual connections (VC's).

We consider two practical organizations of cells: 1) a one-dimensional (linear) array of cells as found alongside a highway, and 2) a two-dimensional arrangement of cells, both as shown in Fig. 1. The tree-VEN is particularly suitable for interconnecting linear and two-dimensional arrangements of access points as shown in Fig. 2 and Fig. 3. Fig. 2 shows a circular array of eight access points connected using a binary tree. The j^{th} switch on the i^{th} level in the tree is denoted by $S_{i,j}$, where the root of the tree is at level 0. Similarly, the j^{th} link on the i^{th} level of the tree is denoted by $L_{i,j}$. A mobile is directly connected to the leaf nodes of tree-VEN (also called access points). Fig. 3 shows an example of a binary tree-VEN interconnecting 64 access points using well known *H-tree*[16] embedding. The underlying ATM network is not shown in these figures.

Now consider the effect of mobility in the wireless portion of the networks on the resources (capacity) required in the links of the tree-based virtual embedded network. For simplicity of analysis, we assume that all the sessions go through the root of the tree. Naturally, if the local communication within a tree-VEN would take the shortest path and not always go through the root of the tree, the overall resources required would be less. Therefore, results obtained in this paper are somewhat pessimistic.

For the example of Fig. 2, if a mobile moves from access points $S_{3,0}$ to $S_{3,1}$, there is no need to negotiate a change of resources on links $L_{0,0}$ and $L_{1,0}$. However, the link $L_{2,1}$ will require more resources and the link $L_{2,0}$ will require fewer resources. This example clearly demonstrates that not all the links between the mobile and the root of tree-VEN require reallocation of resources due to mobility. In fact, the links closer to the access points are more likely to require reallocation of resources due to handoffs compared to the links that are further away.

We study the resource allocation in detail in the rest of the paper, based on the user characteristics and cost criteria defined in the following subsections.

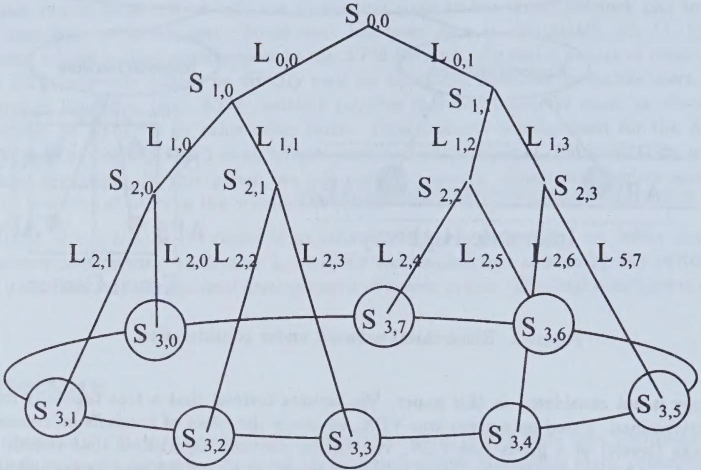

S 0,0
L 0,0
L 0,1
S 1,0
S 1,1
L 1,0
L 1,1
L 1,2
L 1,3
S 2,0
S 2,1
S 2,2
S 2,3
L 2,1
L 2,0
L 2,2
L 2,3
L 2,4
L 2,5
L 2,6
L 5,7
S 3,0
S 3,7
S 3,6
S 3,1
S 3,2
S 3,3
S 3,4
S 3,5

Figure 2: An example of virtual embedded network using tree topology to interconnect a ring of eight access points.

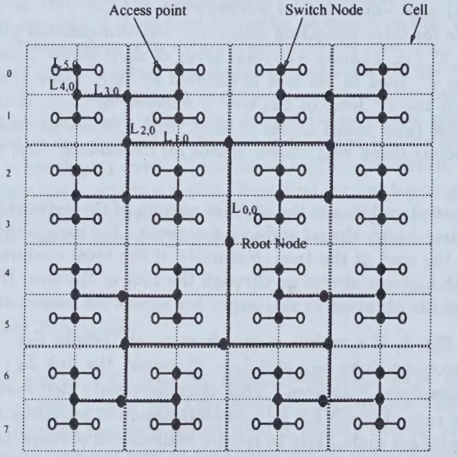

Access point
Switch Node
Cell
L 5,0
L 4,0
L 3,0
L 2,0
L 1,0
L 0,0
Root Node
0
1
2
3
4
5
6
7
0
1
2
3
4
5
6
7

Figure 3: An example of virtual embedded network using tree topology to interconnect a mesh of 64 access points.

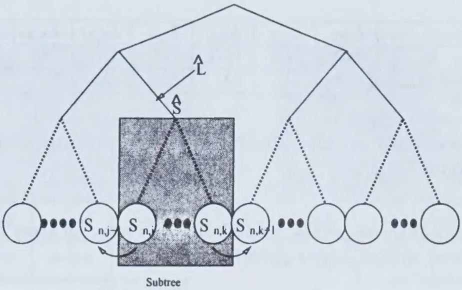


Figure 4: Subtree of tree-VEN constructed over a one-dimensional array of access points.

Therefore, the instantaneous rate of handoffs seen by any link $\hat{L}$ in the tree-VEN is given by

$$\hat{\lambda} = \frac{\lambda}{2}(C_{n,j} + C_{n,k} + C_{n,j-1}, + C_{n,k+1}). \qquad (2)$$

For a homogeneous system in steady state, for a large number of users and access points, the average number of users in all the cells is the same. Therefore, $C_{n,l} \simeq C$ for all the l. Hence, the handoff rate at any link $\hat{L}$ is given by

$$\hat{\lambda} = 2C\lambda \qquad (3)$$

In other words, the number of handoffs seen by all links of the tree-VEN is the same even though the number of users served by different links may be significantly different. Thus, the average processing requirement on different links in the tree to process handoffs is identical regardless of the number of users served by that link. Observe that the rate of handoffs seen by a switch is given by the sum of the handoffs seen by the down-links connecting the switch. Therefore, in case of binary tree, the rate of handoffs seen by the switch $\hat{S}$ is given by $2\hat{\lambda}$

3.2 Two-Dimensional Array of Access Points

In this section, we consider a two-dimensional array of access points connected by a binary tree-VEN. Well-known H-tree embedding is used for connecting a binary tree to the mesh of access points. Each cell of the array is square and hence has four neighbors. Since a mobile user moves to one of the four neighbors with equal probability, the handoff rate per user at every cell boundary is given by $\frac{\lambda}{4}$. Let the two-dimentional array of access points be connected by a balanced tree-VEN of degree r and depth $n+1$. The j^{th} switch on the i^{th} level in the tree is denoted by $S_{i,j}$, where the root of the tree is at level 0. Similarly, the j^{th} link on the i^{th} level of the tree is denoted by $L_{i,j}$. The number of active users served by link $L_{i,j}$ is denoted by $C_{i,j}$.

Consider an example of an 8×8 mesh of access points connected to a single tree-VEN as shown in Fig. 3. The depth of the binary tree-VEN for this example is 6. The link $L_{0,0}$ serves half of the access points, an 8×4 array of access points, as shown in Fig. 5. When a user crosses (moves out or moves in) any of the outer edges of this array, the link $L_{0,0}$ will see a handoff. Since, on average, all the cells serve equal numbers of users, the rate of handoff seen by the link $L_{0,0}$ is given by:

$$\lambda_{0,0} = \frac{\lambda}{4} \times 2 \times (8 + 8 + 4 + 4) = 12\lambda \qquad (4)$$

Link	$L_{0,k}$	$L_{1,k}$	$L_{2,k}$	$L_{3,k}$	$L_{4,k}$	$L_{5,k}$	$L_{6,k}$
Users	$64N$	$32N$	$16N$	$8N$	$4N$	$2N$	N
handoff Rate	12λ	8λ	6λ	4λ	3λ	2λ	1λ

Table 1: The number of users served and the rate of handoffs seen by links at different levels of tree-VEN for 8 × 8 mesh of access points.

In general, for an $N \times N$ array of access points, the link $L_{0,k}$ will serve an array of $N \times \frac{N}{2}$ access points and therefore, the handoff rate seen by link $L_{0,k}$ is given by:

$$\lambda_{0,k} = \frac{\lambda}{4} \times 2 \times (2N + 2 \times \frac{N}{2}) = \frac{3\lambda N}{2} \tag{5}$$

The link $L_{1,0}$ serves an array of 4 × 4 cells as shown in Fig. 5. Therefore, the rate of handoff seen by it is given by:

$$\lambda_{1,0} = \frac{\lambda}{4} \times 2 \times (4+4+4+4) = 8\lambda \tag{6}$$

In the general case, the link $L_{1,k}$ will serve a $\frac{N}{2} \times \frac{N}{2}$ array of access points. Therefore,

$$\lambda_{1,k} = \frac{\lambda}{4} 2 \times (2 \times \frac{N}{2} + 2 \times \frac{N}{2}) = \lambda N \tag{7}$$

In general, for any link $L_{i,k}$ of the tree-VEN, the handoff rate is given by:

$$\lambda_{i,k} = \frac{\lambda}{2}(\frac{N}{2^i} + \frac{N}{2^{\lceil \frac{i+1}{2} \rceil}}) \tag{8}$$

Unlike the linear array of cells, the handoff rate is not identical for all the links on the tree-VEN connecting mesh of access points. In fact, the rate of handoffs for the links increases as the level increases. However, the increase in the number of users served by a link at a higher level of the tree is significantly greater than the increase in the rate of handoff. For the example of the 8 × 8 mesh of access points, the number of users served by links and the respective rate of handoffs are summarized in Table 1. It is observed that by deploying a tree-VEN, the load for processing handoffs in terms of resource reallocation can be significantly reduced compared to the basic scheme where an old connection is terminated and a new connection is established for every handoff. It is clear that the number of users served by a link doubles as we move up in the tree, while the number of handoffs does not increase at the same rate.

4 Conclusions

We have proposed a tree-based Virtual Embedded Network architecture to address resource reallocation problem for mobile users connected to an arbitrary topology ATM backbone network. The approach helps us significantly reduce the resource reallocation rate on different links of the ATM networks for both linear and two-dimensional arrangements of access points. We show that for a linear array of access points, the resource reallocation rate for any link of VEN (and thus a corresponding link of the ATM network) is the same for a homogeneous system even though the number of mobile users served by links of the tree

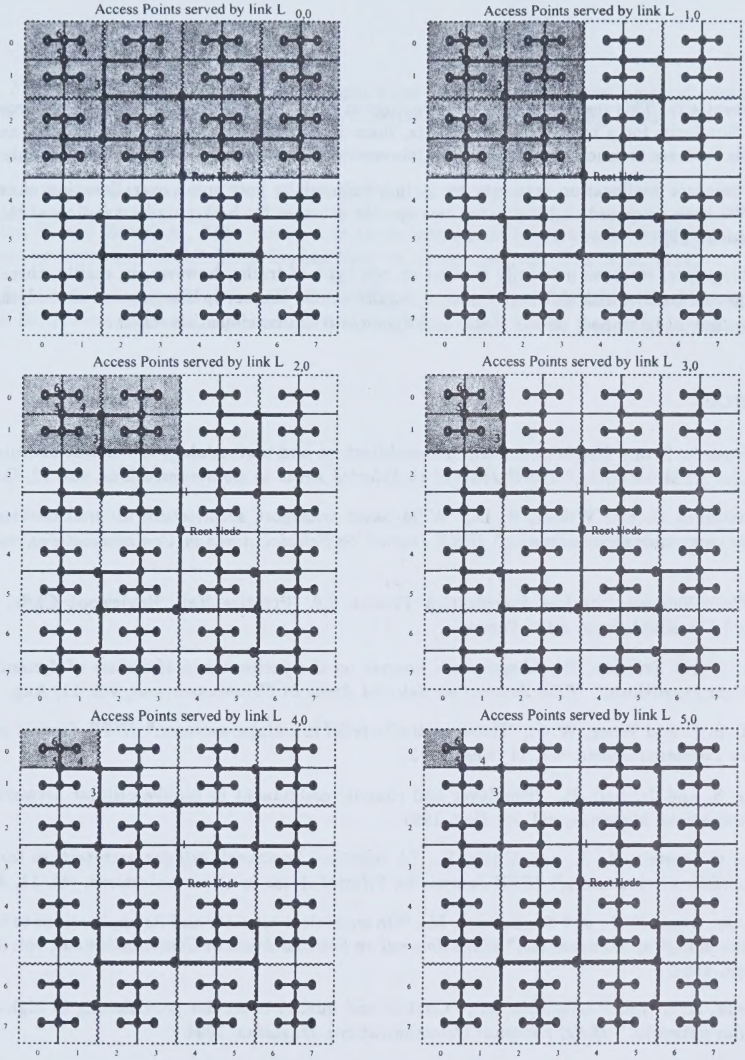

Access Points served by link L $_{0,0}$
Access Points served by link L $_{1,0}$
Access Points served by link L $_{2,0}$
Access Points served by link L $_{3,0}$
Access Points served by link L $_{4,0}$
Access Points served by link L $_{5,0}$
Root Node
0
1
2
3
4
5
6
7

Figure 5: Example of 8 × 8 mesh of access points connected by binary tree-VEN using H-tree layout.

grows exponentially. This results in significant savings in ATM connection management resources and cost of service. Similarly, for a mesh of access points, there is a modest increase in the resource reallocation; however, this increase is much smaller than the increase in the number of users served by a link.

The resource reallocation rate may be further reduced by very small overallocation of resources as shown in. We have proposed and evaluated two specific schemes for better understanding of this trade-off are presented in [17].

In this paper, we have primarily focused on one type of traffic; however, in reality, there are many different types of traffics with different resource requirements. We are in the process of studying different traffic classes and their impact on the static and dynamic resource allocation scheme.

References

[1] Acampora, A. S. and Naghshineh, M., "An architecture and methodology for mobile-executed handoff in cellular ATM networks," *IEEE Journal on Selected Areas in Communications*, vol. 12, Oct. 1994.

[2] Raychaudhuri, D. and Wilson, N. D., "ATM-based transport architecture for multiservices wireless personal communication networks," *IEEE Journal on Selected Areas in Communications*, vol. 12, Oct. 1994.

[3] *ATM User Network Interface Specification Version 3.0.* Prentice Hall, Englewook Cliffs, NJ, 1993. Version 3.1 available from ATM Forum.

[4] Zander, J. and Eriksson, H., "Asymptotic bounds on the performance of a class of dynamic channel assignment algorithms," *IEEE Journal on Selected Areas in Communications*, vol. 11, Aug. 1993.

[5] Yum, T.-S. P. and Wong, W.-S., "Hot-spot traffic relief in cellular systems," *IEEE Journal on Selected Areas in Communications*, vol. 11, Aug. 1993.

[6] Tekinay, S. and Jabbari, B., "Handover and channel assignment in mobile cellular networks," *IEEE Communications Magazine*, vol. 29, Nov. 1991.

[7] I, C.-L., Greenstein, L. J., and Gitlin, R., "A microcell/macrocell cellular architecture for low- and high-mobility wireless users," *IEEE Journal on Selected Areas in Communications*, vol. 11, Aug. 1993.

[8] Guerin, R., Ahmadi, H., and Naghshineh, M., "On equivalent capacity and its applications to bandwidth allocation in high speed networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 968–981, Sept. 1991.

[9] Acampora, A. S. and Naghshineh, M., "Control and quality-of-service provisioning in high-speed microcellular networks," *IEEE Personal Communications Magazine*, 1994.

[10] Pollini, G., Meier-Hellstern, K., and Goodman, D. J., "Handover protocols between metropolitan area networks," in *Proceedings of the 1992 GLOBECOM*, vol. 1, pp. 11–15, 1992.

[11] Amitay, N., "Distributed switching and control with fast resource assignment/handoff for personal communication systems," *IEEE Journal on Selected Areas in Communications*, vol. 11, pp. 842–849, May 1993.

[12] Cohen, R., Patel, B., and Segall, A., "Handover in a micro-cell packet switched mobile network," in *Proceedings of the 1995 IEEE INFOCOM*, 1995. Also, IBM Research report, RC-19432, 1994.

[13] Wang, J. Z., "A fully distributed location registration strategy for universal personal communication systems," *IEEE Journal on Selected Areas in Communications*, vol. 11, Aug. 1993.

[14] Perlman, R., *Interconnections: bridges and routers.* Addison-Wesley, 1992.

[15] Bertsekas, D. and Gallager, R., *Data Networks.* Prentice Hall, 1987.

[16] Horowitz, E. and Zorat, A., "The binary tree as an interconnection network: Applications to multiprocessor systems and VLSI," *IEEE Transactions on Computers*, vol. C-30, Apr. 1981.

[17] Patel, B. V. and Schwartz, M., "Impact of mobility on atm backbone networks," in *Proceedings of the Seventh IEEE International Conference on Personal, Indoor, and Mobile Radio Communications*, 1995.

The Loss Process of Messages with Redundant Packets in an M/M/1/K Queue

Omar Ait Hellal, Eitan Altman and Alain Jean-Marie
INRIA Sophia Antipolis, B.P. 93
06902 Sophia Antipolis, France

Abstract

The purpose of this paper is to study the loss probabilities of messages in an M/M/1/k queue when redundant packets are added. The input to the queue is generated by several sessions, and is modeled by a superposition of Poisson inputs. We use multi-dimensional probability generating functions for solving the recursions introduced in Cidon, Khamisy and Sidi [2] for computing the loss probabilities and derive analytical formulas. Using asymptotic analysis (for large n and small ρ), we obtain good approximations with very low complexity of computation and memory. Our analysis allows us to investigate when does adding redundancy decrease the loss probabilities.

1 Introduction

The loss probabilities of packets in queuing networks is an important performance measure in telecommunication networks and some other applications. Rapid progress in the development of fiber optics allows to achieve a bit error rate of 10^{-14}; information loss is then essentially due to congestioned nodes and buffer overflow.

Often, when messages are divided in several packets, the loss of one packet results in the loss of the whole message. In order to reduce the loss probabilities, one may add redundant packets, so that lost packets can be often reconstructed. Indeed, there exist erasure recovery codes that, by adding k redundant packets to a message, enable to reconstruct up to k losses (see [2,5,6] and references therein). We note, however, that by adding redundant packets, the workload increases and thus the loss probability of a packet increases.

We analyse the tradeoff between the effects of increase of workload and the recovery of lost packets, and calculate the probability of no more losses than k packets within n consecutive ones in the presence of k redundant packets. The computations are based on recursive formulas obtained by Cidon, Khamisy and Sidi [2].

In [5], the authors have used an approximation based on an assumption of independence between consecutive losses, and shown that redundancy results in decrease of loss probabilities by 10 to 100. Exact numerical methods based on recursions [2] led to an opposite conclusion, i.e. that redundancy causes increase in loss probabilities. One of the advantages of our analytical approach, together with the asymptotic approximations which we present, is that they enable to study both qualitative and quantitative behavior of the effect of redundancy in a systematic way. As was already shown in [3] for the case of a single source, we show that for both light traffic as well as heavy traffic conditions, redundancy decreases loss probabilities.

The paper is structured as follows: in section 2, we describe the model and calculate the probability generating function. The asymptotic analysis is presented in section 3. In section 4, we show that in light traffic, adding redundancy decreases the loss probabilities. Numerical examples which illustrate this improvement are given in section 5.

2 The Model and the Main Results

We consider an M/M/1 queue with a finite buffer of size M. We assume that packets arrive to the queue from S independent sources, i.e. the inter-arrival times and the transmission times of packets from each source are mutually independent. The arrival process from source s, $s = 1, 2, \cdots, S$, is assumed to be Poisson with rate λ_s. The overall arrival process to the system is then Poisson with rate $\lambda \triangleq \sum_s^S \lambda_s$. Define $\rho = \lambda/\mu$, $\rho_s = \lambda_s/\mu = p_s\rho$. We summarize the recursive sheme introduced in [2] for computing $P^s(j,n), s = 1,2,..,S$ which are the probabilities of j losses among n consecutive ones originating from source s. For the system with Poisson arrivals with rate λ and exponential transmission rate μ, in steady state, the probability of finding i packets in the system at an arbitrary epoch is given by

$$\Pi(i) = \frac{\rho^i}{\sum_{l=0}^M \rho^l} \tag{1}$$

Define $Q_i(k)$ to be the probability that k packets out of i leave the system durant an inter-arrival epoch. We have

$$\begin{aligned} Q_i(k) &= \rho\alpha^{k+1} \qquad 0 \le k \le i-1 \\ Q_i(i) &= \alpha^i, \qquad \text{where} \quad \alpha := (1+\rho)^{-1} \end{aligned} \tag{2}$$

Denote by $P_i^{s,a}(j,n)$ *resp* $P_i^{\bar{s},a}(j,n)$, $i = 0,1,...,M$, $s = 1,2,...,S$, $n \ge 1$, $0 \le j \le n$, the probabilities of j losses in a block of n packets coming from source s, given that there are i packets in the system just before the arrival of the first packet in the block, and just before the arrival of a packet from any other source (denoted by $\bar{s}$), respectively. Since the first packet in the block is arbitrary, we have

$$P^s(j,n) = \sum_{i=0}^M \Pi(i) P_i^{s,a}(j,n) \tag{3}$$

The probability that an arbitrary arrival is from source s is equal to λ_s/λ. Define $p_s \triangleq \lambda_s/\lambda$ and $p_{\bar{s}} \triangleq 1 - p_s$, the recursive sheme is then

$$P_i^{s,a}(j,1) = \begin{cases} 1 & j = 0 \\ 0 & j \geq 1, \end{cases} \qquad i = 0, 1, \ldots, M-1 \tag{4}$$

$$P_M^{s,a}(j,1) = \begin{cases} 1 & j = 1 \\ 0 & j = 0, j \geq 2, \end{cases} \tag{5}$$

For $n \geq 2$ we have

$$\begin{aligned} P_i^{s,a}(j,n) &= \sum_{k=0}^{i+1} Q_{i+1}(k) \left[p_s P_{i+1-k}^{s,a}(j,n-1) + p_{\bar{s}} P_{i+1-k}^{\bar{s},a}(j,n-1) \right] \qquad 0 \leq i \leq M-1 \\ P_M^{s,a}(j,n) &= \sum_{k=0}^{M} Q_M(k) \left[p_s P_{M-k}^{s,a}(j-1,n-1) + p_{\bar{s}} P_{M-k}^{\bar{s},a}(j-1,n-1) \right] \end{aligned} \tag{6}$$

where $P_i^{\bar{s},a}(j,n)$ for $n \geq 1$ is given by:

$$\begin{aligned} P_i^{\bar{s},a}(j,n) &= \sum_{k=0}^{i+1} Q_{i+1}(k) \left[p_s P_{i+1-k}^{s,a}(j,n) + p_{\bar{s}} P_{i+1-k}^{\bar{s},a}(j,n) \right] \qquad 0 \leq i \leq M-1 \\ P_M^{\bar{s},a}(j,n) &= P_{M-1}^{\bar{s},a}(j,n) \end{aligned} \tag{7}$$

The complexity of these recursions is $O(M^2nj)$ in arithmetic operations and $O(M^2)$ in memory space.

Next, we state the main results, whose detailled proofs are given in [1]. Define

$$q_s(y,z) \stackrel{def}{=} \sum_{j=0}^{\infty} \sum_{n=1}^{\infty} y^j z^{n-1} P^s(j,n)$$

Let $x_1(z)$ and $x_2(z)$ be the solutions in x of

$$x^2 - (1+\rho)x + \rho(p_{\bar{s}} + p_s z) = 0$$

i.e.

$$x_1(z) = \frac{1 + \rho + \sqrt{(1+\rho)^2 - 4\rho(p_{\bar{s}} + p_s z)}}{2}, \qquad x_2(z) = \frac{1 + \rho - \sqrt{(1+\rho)^2 - 4\rho(p_{\bar{s}} + p_s z)}}{2}$$

We shall often write simply x_1 and x_2 for $x_1(z)$ and $x_2(z)$. Both these functions are analytic in the disk $\{|z| < ((1+\rho)^2 - 4\rho_{\bar{s}})/4\rho_s\}$. Define, for all k, $\delta_k = x_1^k - x_2^k$, $\phi_k = (p_{\bar{s}} + p_s z)\delta_{k-1} - \delta_k$. Let $R_M = (\sum_{l=0}^{M} \rho^l)^{-1}$.

Proposition 2.1. *The probability generating function q_s is given by*

$$q_s(y,z) = \frac{R_M}{(1-z)} \left(R_M^{-1} + \frac{\rho^{M-1}\delta_{M+1}}{z\phi_M} + \frac{\rho^{M-1}(1-z)\left[\delta_{M+1}\right]^2}{z\phi_M} \left[\frac{1}{z\delta_M - \delta_{M+1} - \rho z \phi_M y} \right] \right) \tag{8}$$

Once the probability generating function is obtained by Proposition 2.1, one can obtain the required probabilities by inverting q_s. We focus in the sequel on $P_\rho^s(>j,n)$, which is the probability of losing more than j packets out of n consecutive ones coming from source s. We investigate in particular the case of $j = 0, 1$, in order to be able to decide when does adding redundancy packet to each message result in decrease of the loss probability.

We shall use the notation $\left[z^k\right] f(z)$ to denote the coefficient of z^k in the Taylor expansion of the function $f(z)$, i.e. if $f(z) = \sum_k f_k z^k$ then $\left[z^k\right] f(z) = f_k$.

Corollary 2.1. *The probability of losing more than j packets out of n consecutive packets that arrive from source s is*

$$P_\rho^s(>j,n) = R_M \rho^{M+j} \left[z^{n-1-j}\right] \frac{1}{z-1} \left[\frac{\delta_{M+1}}{z\delta_M - \delta_{M+1}} \right] \left(\frac{\phi_M}{z\delta_M - \delta_{M+1}} \right)^j \tag{9}$$

In the following, we obtain a simple recursion on n, for computing the probabilities $P_\rho^s(>0,n)$. Thus, we avoid a recursion on j and a resolution of a set of linear equations of size M for all j and all n required in [2].

Define β, γ and θ as

$$\gamma = 1+\rho, \quad \beta = \sqrt{(1+\rho)^2 - 4\rho_{\bar{s}}}, \quad \theta = \frac{4\rho_s}{(1+\rho)^2 - 4\rho_{\bar{s}}} \tag{10}$$

Let

$$G = \begin{cases} \frac{M}{2} - 1 & \text{for M even} \\ \frac{M-1}{2} & \text{otherwise} \end{cases} \qquad a_n = \begin{cases} (-\theta)^n \sum\limits_{k=n}^{G+1} \binom{M+1}{2k+1} \binom{k}{n} \beta \left(\frac{\beta}{\gamma}\right)^{2k} & n \le G+1 \\ 0 & \text{otherwise} \end{cases}$$

$$b_n = \begin{cases} (-\theta)^{n-1} \sum\limits_{k=n}^{G+1} \binom{M+1}{2k+1} \binom{k}{n} \left(\frac{4\gamma\,(2k+1)\,n}{\beta\,(M+1)\,(M+1-2k)} + \theta\beta \right) \left(\frac{\beta}{\gamma}\right)^{2k} & n \le G+1 \\ 0 & \text{otherwise} \end{cases}$$

Corollary 2.2. *For $n \ge 1$ we have*

$$P_\rho^s(>0,n) = R_M \rho^M + \frac{1}{a_0} \sum_{k=1}^{n-1} \left(b_{n-k} P_\rho^s(>0,k) + R_M \rho^M a_k \right) \tag{11}$$

We see that for $j = 0$, we obtain a procedure for computing the loss probabilities, of complexity $O(Mn+n^2)$ in operations and $O(M^2)$ in memory space if we consider that all the values $\begin{pmatrix} M \\ r \end{pmatrix}$, $0 \leq r \leq M$ remain in memory. This is a substantial improvement over the complexity of $O(M^2n)$ in operations and $O(M^2)$ in memory space required in [2].

In case $j > 0$, we obtain a procedure whose complexity is of the order $O(j^2n^2 + Mnj^2)$ in operations and $O(M^2j^2)$ in memory space [1].

A detailed proof of Proposition 2.1 and the proofs of Corollary 1 and 2 are given in [1].

3 Asymptotic behavior

In order to reduce the complexity of calculations of $P_\rho^s(> j, n)$, we make the following approximations, based on an asymptotic expression for these probabilities for n large. From (9) we have

$$\begin{aligned} P_\rho^s(> j,n) &= -R_M\rho^M \left[z^{n-1}\right] \frac{1}{1-z} \frac{x_1^{M+1}}{x_1^M(z-x_1)} \left(\frac{zx_1^M(1-x_1)}{x_1^M(z-x_1)}\right)^j + \left[z^{n-1}\right] f_j(z) \\ &= -R_M\rho^M \left[z^{n-1}\right] \frac{1}{1-z} \left(\frac{\rho_{\bar{s}}}{x_1-\rho_{\bar{s}}} + \frac{1}{1-x_1}\right) \left(x_2 - \frac{\rho_{\bar{s}}(x_1-\rho)}{x_1-\rho_{\bar{s}}}\right)^j + \left[z^{n-1}\right] f_j(z) \\ &= \tilde{p}_\rho^s(> j,n) + \left[z^{n-1}\right] f_j(z). \end{aligned}$$

It can be shown that the term $[z^{n-1}]\, f_j(z)$ can be neglected for large n and $n < M+1$ and hence $P_\rho^s(> j,n) \cong \tilde{P}_\rho^s(> j,n)$, which we compute next.

Proposition 3.1. *For $n \geq 1$ we have*

$$\tilde{P}_\rho^s(> 0,n) = \begin{cases} \dfrac{R_M\rho^M}{1-\rho_{\bar{s}}} \left[(1-\rho)n + \left(\dfrac{\rho_s}{1-\rho} - \dfrac{\rho_s\rho_{\bar{s}}}{1-\rho_{\bar{s}}}\right) + \theta^n O\left(n^{-3/2}\right)\right] & \text{if } \rho < 1 \\ \dfrac{1}{M+1} \dfrac{1}{p_s} \left[\left(2\sqrt{p_s} + \dfrac{p_{\bar{s}}}{\sqrt{p_s} n}\right) \dfrac{\sqrt{n}}{\sqrt{\pi}} \left(1 + O\left(\dfrac{1}{n}\right)\right) - p_{\bar{s}}\right] & \text{if } \rho = 1 \\ 1 - \left(\dfrac{4\rho_s^2}{\rho(\rho-1)^3} - \dfrac{\rho_s\rho_{\bar{s}}}{\rho(1-\rho_{\bar{s}})} \left(\dfrac{1}{\rho-1} - \dfrac{\rho-1}{(1+\rho_s-\rho_{\bar{s}})^2}\right)\right) & \\ \theta^{n-1} \dfrac{\beta\, n^{-3/2}}{(1-\rho_{\bar{s}})\sqrt{\pi}} (1 + o(1)) & \text{if } \rho > 1 \end{cases} \tag{12}$$

$$\tilde{P}^s_\rho(>1,n) = \begin{cases} \dfrac{R_M \rho^M \rho_s}{(1-\rho_{\bar{s}})^2}\left[(1-\rho)n + \dfrac{\rho_s+\rho-1}{1-\rho} - \dfrac{\rho_s\rho_{\bar{s}}}{1-\rho_{\bar{s}}} + \theta^n O\left(n^{-3/2}\right)\right] & \textit{if } \rho < 1 \\ \dfrac{1}{M+1}\dfrac{1}{p_s}\left[\left(2\sqrt{p_s} + \dfrac{2p_{\bar{s}}}{\sqrt{p_s}n}\right)\dfrac{\sqrt{n}}{\sqrt{\pi}}\left(1+O\left(\dfrac{1}{n}\right)\right) - (1+p_{\bar{s}})\right] & \textit{if } \rho = 1 \\ 1 - \dfrac{\rho_s}{1-\rho_{\bar{s}}}\biggl(\dfrac{4\rho_s^2}{\rho(\rho-1)^3} + \dfrac{\rho_{\bar{s}}(\rho-1)}{\rho(1-\rho_{\bar{s}})(1+\rho_s-\rho_{\bar{s}})}\biggl(\dfrac{(1+\rho_s-\rho_{\bar{s}})^2}{(\rho-1)^2} & \\ -\dfrac{\rho_s}{(1+\rho_s-\rho_{\bar{s}})} + \rho_{\bar{s}}^2 + \dfrac{4\rho_s\rho_{\bar{s}}^2(1-\rho_{\bar{s}})}{(1+\rho_s-\rho_{\bar{s}})^2}\biggr)\biggr)\dfrac{\theta^{n-1}\beta\, n^{-3/2}}{(1-\rho_{\bar{s}})\sqrt{\pi}}\,(1+o(1)) & \textit{if } \rho > 1 \end{cases} \tag{13}$$

The proof of this proposition is given in [1] and is based on [4].

Proposition 3.2. *For $n \geq 1$, $0 \leq j < n$, we have for small ρ*

$$P^s_\rho(>j,n) = \rho^M \rho_s^j(n-j+O(\rho_s)).$$

4 When is it better to add redundancy

In this section we compare the loss probabilities of messages of size n, with and without an additional redundancy of a single packet. Without loss of generality, we may rescale time so that the service rate is one: $\mu = 1$. We assume that the rate at which messages arrive is the same for the two cases and is given by $x = p_{\bar{s}}x + p_s x = x' + x''$. We ditinguish two cases:

1. adding redundancy for all sources. Hence, the rate at which packets arrive is $\lambda = \rho = (n+1)x$.
2. adding redundancy only for source s; the workload is then $\lambda = \rho = nx + x''$ and $\rho_{\bar{s}}$ still constant.

The message is lost, in both last cases, if and only if more than one packet is lost out of $n+1$ consecutive ones coming from source s.

We are thus interested in the difference $\Delta = P^s_{nx}(>0,n) - P^s_{(n+1)x}(>1,n+1)$. If $\Delta > 0$ then the redundancy decreases the loss probabilities of messages.

Proposition 4.1. *For any n and M, adding redundancy results in a decrease of the loss probabilities for all x small enough (light traffic regime).*

Proof: We consider case 1. Case 2 follows similarly. From Proposition 3.2 we have $P^s_{nx}(>0,n) = (nx)^M(n+O(nx''))$ and

$$P^s_{(n+1)x}(>1,n+1) = ((n+1)x)^{M+1}p_s(n+O(nx'')).$$

The proof now follows by noting that

$$\lim_{x\to 0}\frac{P^s_{(n+1)x}(>1,n+1)}{P^s_{nx}(>0,n)} = \lim_{x\to 0}\left(\frac{n+1}{n}\right)^M (n+1)p_s x = 0 < 1. \tag{14}$$

5 Numerical examples

We have shown that adding redundancy is profitable in light traffic. A natural question is how small should the traffic be in order for this conclusion to hold in practice.

Below, we fix ρ, p_s and obtain a set of n and M for which redundancy will lead to better performance and for which the loss probability of messages is of a given order (ex:10^{-8}). We shall restrict to a family of n and M that are inter-related by $n \cong \eta M$, where η is a constant to be determined, and we shall consider $n >> 1$. In fact, the approximations turn out to be quite accurate even for moderate values of n and M. In Figure 1, we display the values of $P^s_\rho(>0, n)$ and its approximations (from Proposition 3.1):

$$A_0(>0,n) \overset{\text{def}}{=} \frac{R_M \rho^M}{(1-\rho_{\bar{s}})}\left((1-\rho)n + \rho_s\left(\frac{1}{1-\rho} - \frac{\rho_{\bar{s}}}{1-\rho_{\bar{s}}}\right)\right),\ B_0(>0,n) \overset{\text{def}}{=} \frac{R_M \rho^M}{(1-\rho_{\bar{s}})}(1-\rho)n$$

in the case $M = 10$, $n \leq 10$, $\rho = 0.4$ and $p_s = 0.6$. In Figure 2, we make the same comparison for $P^s_\rho(>1, n)$ and

$$A_1(>1,n) \overset{\text{def}}{=} \frac{R_M \rho^M \rho_s}{(1-\rho_{\bar{s}})^2}\left((1-\rho)n + \frac{\rho+\rho_s-1}{1-\rho} - \frac{\rho_s\rho_{\bar{s}}}{1-\rho_{\bar{s}}}\right),\ B_1(>1,n) \overset{\text{def}}{=} \frac{R_M \rho^M \rho_s}{(1-\rho_{\bar{s}})^2}(1-\rho)n$$

These approximations are obtained from Proposition 3.1($\rho < 1$), by taking the two first and the first term, respectively, in the asymptotic expansion (in n) of $\tilde{P}^s$.

5.1 Adding redundancy for all sources

We wish to determine x^* for which $P^s_{(n+1)x}(>1, n+1) - x^* P^s_{nx}(>0, n) = 0$. $[0, x^*]$ will be the interval for which we should use redundancy. From (14), we have for large M

$$x^* = \frac{1}{p_s(n+1)}\left(\frac{n}{n+1}\right)^M \cong \frac{1}{p_s \eta M}\frac{n}{n+1}\left(1 - \frac{1}{\eta M}\right)^M \cong \frac{n}{n+1}\frac{\exp(-1/\eta)}{p_s \eta M}$$

Since $\rho^*_s = p_s x^* \eta M$, we have

$$\rho^*_s \cong \frac{n}{n+1}\exp(-1/\eta) \cong \exp(-1/\eta) \tag{15}$$

More generaly, for $\rho_j = \frac{n}{n+j}\rho^*$ adding j packets leads to better performance. This heuristic is quite optimistic and we obtain this experimental result: We fix $\rho = 0.6$ and for $p_s \in]0, p^*_s]$ where $p^*_s = exp(-1/\eta)$, adding one redundant packet decreases the loss probabilities of messages and for the same values of p_s, for $\rho = \frac{n}{n+j}0.6$, adding j packets decreases these probabilities.

Example 1: Let $\rho = 0.6$, $p_s = 0.1$, we wish to determine n and M for which the loss probability is of order 10^{-8} and redundancy leads to better performance. It folows from (15) that $\eta \geq 0.36$, from Proposition 3.1, we have

$$P^s_{0.6}(>0, n) \cong \frac{1-0.6}{1-0.6^{M+1}}\frac{0.6^M}{1-0.54}(1-0.6)0.36M \cong 10^{-8} \quad \Rightarrow M \cong 39,\ n = \eta M \cong 15$$

The exact calculation for $A(\rho) \overset{\text{def}}{=} P^s_{nx}(> 0, n)$ and $S(\rho) \overset{\text{def}}{=} P^s_{(n+1)x}(> 1, n+1)$ yields that for $n = 15$, $A(0.6) = 1.17*10^{-8}$ and $S(0.6) = 1.88*10^{-8}$. For $n = 19$, $S(0.6) = 1.42*10^{-8}$ et $A(0,6) = 1.48 * 10^{-8}$. We have to take η greater than the value $\eta = 0.36$ obained above. If we choose $\eta = 0.5$ we obtain $M = 40$ and $n = 20$; $A(0.6) = 9.39 * 10^{-9}$ and $S(0.6) = 8.59 * 10^{-9}$. In Figure 3, we display the probability with and without redundancy as a function of p_s for $\rho = 0.6$, $M = 39$ and $n = 19$. We note that for the sources whose proportion in the overall arrival stream does not exceed 10% ($p_s \leq 0.1$), redundancy decreases their loss probabilities, but not for the others.

We note also, that the loss probability without redundancy decreases in p_s contrary to the loss probability with redundancy which increases in p_s. Figure 4 illustrates the loss probabilities as function of the number of redundant packets for $\rho = (19/(19+4))0.6 \cong 0.5$, $p_s = 0.1, n = 19, M = 39$.

5.2 Adding redundancy only for source s

We are interested by case when redundancy is added only for the source s. We assume that the rate at which messages arrive is the same with and without redundancy. When one redundant packet is added for source s, we have $\lambda = \rho = nx + p_s x$. We proceed similarly as above, we get for large M

$$\rho^*_s = \exp(-p_s/\eta). \tag{16}$$

Remark 5.1. *In all the paper, by substituting 1 for p_s, we obtain all the equations and formulas of [3], the case of single source.*

Exemple 2: Let $\rho = 0.8$ and $p_s = 0.1$, we wish to determine n and M for which the loss probability is of order 10^{-9} and redundancy leads to better performance. From (16) we have $\eta = 0.04$ and from Proposition 3.1 we have

$$P^s_{0.8}(> 0, n) \cong \frac{1-0.8}{1-0.8^{M+1}} \frac{0.8^M}{1-0.72}(1-0.8)0.04M \cong 10^{-9} \quad \Rightarrow M \cong 90,\ n = \eta M \cong 4$$

exact calculation for $A(\rho) \overset{\text{def}}{=} P^s_{nx}(> 0, n)$ and $S'(\rho) \overset{\text{def}}{=} P^s_{nx+p_s x}(> 1, n+1)$ yields that for $n = 4$ and $M = 90$, $S(0.8) = 2.67 * 10^{-9}$ and $A(\rho) = 1.28 * 10^{-9}$ and for $n = 7$, $M = 90$ we have $S(0.8) = 1.86 * 10^{-9}$ and $A(\rho) = 2.12 * 10^{-9}$, thus for $n \geq 7$, $S(\rho) < A(\rho)$. We display in Figure 5 the loss probabilities as function of the number of redundant packets for $n = 8$, $M = 90$, $\rho = 0.8$ and $p_s = 0.1$.

We note that the larger size of messages leads to better perfomance in the presence of redundancy. For the same example, by taking $n = 14$, we reduce the loss probabilities of messages by an order of 10 when we add 4 redundant packets only for source s (Figure 6). (Note that this improvement has negative effect on the other sources.)

For the example above, if we consider only two surces s and $\bar{s}$, we have $p_{\bar{s}} = 0.9$ and $P^{\bar{s}}_{0.8} = 2.114\ 10^{-9}$. When we add 4 redundant packets for s the workload becomes mail$\rho = 0.8 + \frac{4}{14}0.08 = 0.8228$, $p_s = \frac{18}{14}0.1 = 0.1285$ and $p_{\bar{s}} = 1 - 0,1285 = 0.8715$. For these values, we obtain $P^{\bar{s}}_{0.8228} = 2.305\ 10^{-8}$. Finaly, we see that the redundancy in source s has an effect of increasing the loss probability for the other source (by the same order of magnitude as the decrease in the loss probabilities of s).

References

[1] O. Ait-Hellal, *Perte de Messages avec Paquets Redondants dans une File M/M/1/K*, Rapport de stage de D.E.A, Universite de Versailles, September, 1995.

[2] I. Cidon, A. Khamisy and M. Sidi, "Analysis of Packet Loss Process in High-Speed Netwoks", *IEEE Trans. IT*, VOL. 39, No. 1, pp. 98-108, 1993.

[3] E. Altman and A. Jean-Marie, "The loss process of messages in an M/M/1/K queue", *INFOCOM'94*, Toronto, Canada, pp. 1191-1198, June, 1994.

[4] P. Flajolet and A. Odlyzko, "Singularity analysis of generating functions", *SIAM J. Disc. Math.*, Vol. 3, No. 2, pp. 216-240, 1990.

[5] N. Shacham and P. McKenney, "Packet Recovery in High-Speed Networks Using Coding and Buffer Management", *INFOCOM '90*, San Fransisco, CA, pp. 124-131, 1990.

[6] N. Cem Oğuz and E. Ayanoğlu, "Performance Analysis of Two-Level Forward Error Correction for Lost Cell Recovery in ATM Networks", *INFOCOM '95*, pp. 6b.3.1-6b.3.10, 1995.

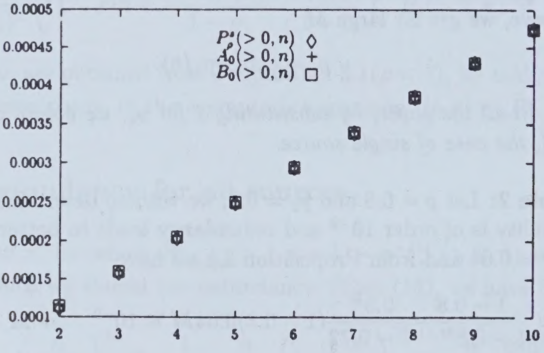


Figure 1: $P^s_\rho(> 0, n)$ and its approximations $\rho = 0.4$, $p_s = 0.6$ and $M = 10$.

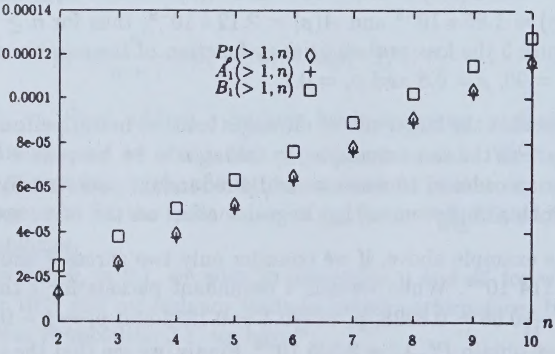


Figure 2: $P^s_\rho(> 1, n)$ and its approximations $\rho = 0.4$, $p_s = 0.6$ and $M = 10$.

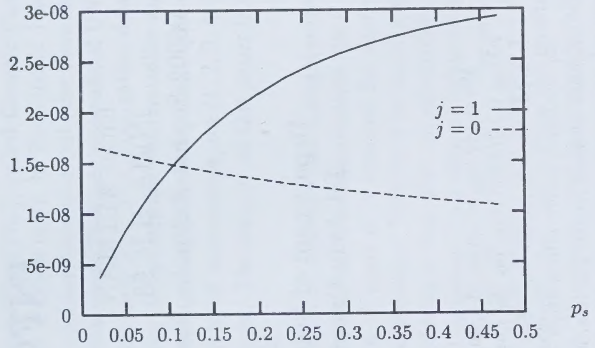


Figure 3: $P^s_\rho(> j, n + j)$, $j = 0, 1$ as function of p_s for $\rho = 0.6$, $n = 19$ and $M = 39$.

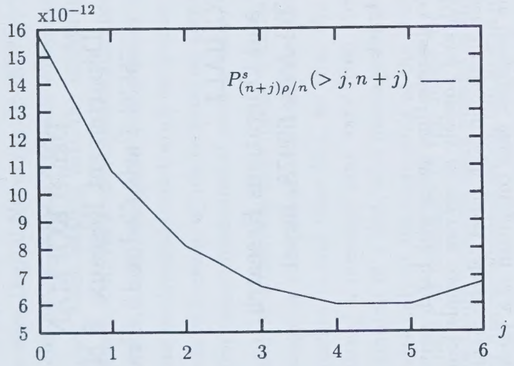


Figure 4: $P^s_{(n+j)\rho/n}(> j, n + j)$ as function of j for $\rho = 0.5$, $p_s = 0.1$, $n = 19$ and $M = 39$

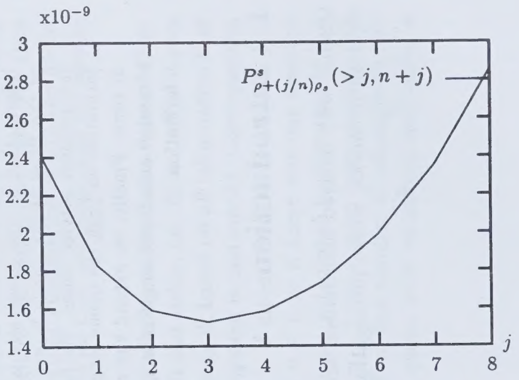


Figure 5: $P^s_{\rho+(j/n)\rho_s}(> j, n + j)$ as function of j for $\rho = 0.8$, $p_s = 0.1$, $n = 8$ and $M = 90$

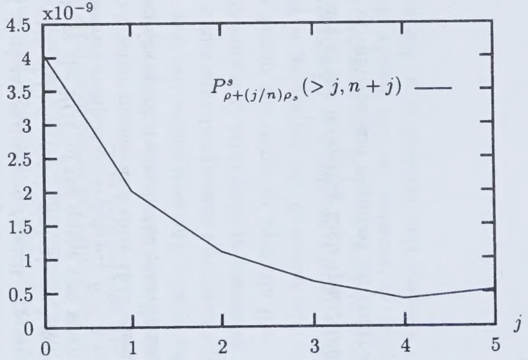


Figure 6: $P^s_{(\rho+(j/n)\rho_s)}(> j, n + j)$ as function of j for $\rho = 0.8$, $p_s = 0.1$, $n = 14$ and $M = 90$

ACCESS CONTROL WITH DELAYED INFORMATION

Eitan ALTMAN
INRIA, B.P. 93
06902 Sophia-Antipolis, France

Daniel KOFMAN
Département Réseaux, E.N.S.T.
75634 Paris Cedex 13, France

Uri YECHIALI
Department of Statistics and Operations Research
Tel-Aviv University, Tel-Aviv 69978, Israel

Abstract

We study the behavior of a single-server discrete-time queue with batch arrivals, where the information on the queue length and possibly on service completions is delayed. Such a model describes situations arrising in high speed telecommunication systems, where information arrives in messages, each comprised of a variable number of fixed-length packets, and it takes one unit of time (a slot) to transmit a packet. Since it is not desirable to attempt service when the system may be empty, we study a model where we assume that service is attempted only if, given the information available to the server, it is certain that there are messages in the queue. We characterize the probability distribution of the number of messages in the queue under some general stationarity assumptions on the arrival process, when information on the queue size is delayed K slots, and derive explicit expressions of the PGF of the queue length for the case of i.i.d. batch arrivals and general independent service times. We further derive the PGF of the queue size when information on both the queue length and service completion is delayed $K = 1$ units of time. Finally, we extend the results to priority queues and show that when all messages are of unit length, the $c\mu$ rule remains optimal even in the case of delayed information.

1 Introduction

Queues with delayed information are quite useful for modeling high-speed telecommunication networks, where propagation delay of information required for controlling the system is nonnegligible with respect to transmission time. A typical example is the following. In Asynchronous Transfer Mode (ATM), several types of customers can use the same User Network Interface (UNI) to access the network. Priority may be assigned to the various types. A field in the Protocol Data Unit (ATM-PDU), called Generic Flow Control (GFC), can be used to control the access at a single UNI. In Broadband Integrated Services Digital Networks (B-ISDN) the transmission time of a cell (the time slot in our model below) equals 0.68μsec when dealing with the 622Mb/sec normalized UNI, and 2.73 μsec when dealing with the 155Mb/sec normalized UNI. The propagation delay is about 4μsec/km. As a consequence, the propagation delay between the controller and the customers is not negligible with respect to the cell transmission time. Such type of applications motivated recent work on queues with delayed information, see [1, 2, 3, 4, 5] and references therein, which focused on control aspects. The purpose of this paper is to study performance measures of such queues.

We analyze a discrete-time queue with batch arrivals of messages and a single server. Each message is comprised of a variable number of packets. The transmission time of a single packet is one time slot, so that the number of time slots required for serving (transmitting) a message may vary from message to message. Service is initiated only at the beginning of a slot, and messages that arrive during a time slot are candidates to begin service only at the beginning of the next slot. The server has a delayed information on the number of messages in the queue, i.e., at the beginning of time slot n, the server knows how many messages were present in the queue at the beginning of time slot $n-K$. The server has no information on the length of any message, nor does it know the number of messages that arrived during any of the last K slots. We assume that it is not desirable to try to serve messages when the system may be empty, since the server may be used for performing other tasks at those times.

We consider two situations regarding information on service completions. In the first (Section 2), the server has full and non delayed information on service completions in previous time slots. For this case we obtain a general decomposition rule assuming that the arrival process is stationary and independent of the service process. We then derive explicit expressions for the Probability Generating Function (PGF) of the queue length when the number of message-arrivals per slot is an i.i.d. sequence, and the service durations are general and i.i.d..

The second situation is when the service completion information is also delayed (Section 4). We restrict the analysis to the case of a unit-slot delay, i.e. $K = 1$. The server will attempt to serve a message at time slot n if and only if, based on all information that the server possesses at that time, it is sure that the system is non empty. Consequently, the server will attempt to serve the queue at time n if either there were at least two messages in the queue at the beginning of time slot $n-1$, or if there was one message at the beginning of time slot $n-1$ but service was not attempted (service would not be attempted at time $n-1$ if, for example, at time $n-2$ the system were empty). We obtain explicit expressions for the steady state PGF of the number of messages in the system by viewing the problem as a discrete time queue with vacations, and using a work decomposition result. Finally (Section 5), we discuss a priority queue with delayed information, where all messages are of unit length. We obtain explicit expressions for the queue size and mean waiting time of a class-m customer, and show that the $c\mu$ rule remains optimal even in the case of delayed information.

2 Delayed information on arrivals

Consider a single server queue, where time is slotted, i.e., time is subdivided into some basic units of length δ; slot j begins at time $j\delta$, where $j = ..., -1, 0, 1, 2, 3, ...$ We consider below batch arrivals of messages into a single queue and we are interested in the process that describes the number of messages in the queue. Each message is composed of a sequence of packets (each with unit size). Thus, the time it takes to transmit a message may vary from message to message. The underlying process is the "late arrival" model described in Section 6.1 of [10]. More precisely, let $\{X_n\}$, $n = -K, -K+1, ..., -1, 0, 1, 2, 3, ...$, be the process of queue lengths, where X_n denotes the number of messages in the queue at the beginning of time slot n. During time-slot n, a batch of A_n messages arrives. Let U_n be a decision variable denoting whether service is attempted at time n ($U_n = 1$), or not ($U_n = 0$). U_n depends on the information available to the server (see below) at the beginning of slot n. We define a sequence $C_n \in \{0, 1\}$ which corresponds to a potential service completion; if service is attempted at the beginning of slot n (and the system is nonempty), then service completion (of a whole message) occurs at the end of the slot if and only if $C_n = 1$. The interpretation of C_n is as follows. Suppose the kth arriving message is of a given length, say $\tau_k \geq 1$ packets, and at time n the server starts to serve that message and continues serving its packets until the whole message is transmitted. Since we assume that it takes one time slot to serve one packet, then $C_m = 0$ for $m = n, n+1, ..., n+\tau_k - 2$, and $C_m = 1$ for $m = n + \tau_k - 1$. Hence, $C_m = 1$ at those time slots at the end of which the service (the transmission) of a whole message is completed.

If $U_n = 1$ and $C_n = 1$, then a message leaves the system. This yields:

$$X_{n+1} = X_n + A_n - U_n C_n \tag{1}$$

The service decisions are governed by the following rule: service is attempted at time n if the server is sure that the system is nonempty at that time. The available information for this decision is the queue length till time $n - K$, as well as the previous decisions U_l and the values of C_l, $l < n$. More precisely, we assume that U_n is given by

$$U_n = 1\left\{X_{n-K} > \sum_{i=1}^{K} U_{n-i} C_{n-i}\right\}. \tag{2}$$

Note, in particular, that if service is initiated at time n, then it will continue till the current message leaves the system. Indeed, assume that $U_n = 1$, and let $m > n$ be the instant at which the message that starts being served at time n is fully transmitted and leaves the queue. Then for any l such that $n < l < m$,

$$X_{l-K} \geq X_{n-K} - \sum_{i=n-K}^{l-K-1} U_i C_i, \tag{3}$$

This follows since $\sum_{i=n-K}^{l-K-1} U_i C_i$ is the number of messages transmitted during time slots $n - K$ to $l - K - 1$. Now,

$$X_{l-K} - \sum_{i=1}^{K} U_{l-i} C_{l-i} = X_{l-K} - \sum_{i=l-K}^{n-1} U_i C_i \geq X_{n-K} - \sum_{i=n-K}^{l-K-1} U_i C_i - \sum_{i=l-K}^{n-1} U_i C_i > 0 \tag{4}$$

The first equality is du to the fact that $\sum_{i=n}^{l-1} U_i C_i = 0$ (no messages leave the system during the time interval $[n, l-1]$) and the last inequality is du to the fact that $U_n = 1$.

Hence $U_l = 1$, $n \leq l \leq m - 1$. Service is thus attempted till the message whose service has started at time n, leaves the system.

Thus, since each message is continuously served until transmission, one may view the system as a discrete time queue with non preemptive server vacations. The vacations are caused by the delay in information which may idle the server even when the system is non-empty. We will use this observation in the sequel.

Next, we make some probabilistic assumptions on the exogenous sequences $\{A_n\}$, and $\{C_n\}$. We consider a probability space on which the two random processes $\{A_n\}$ and $\{C_n\}$ are defined, where $n = -K, -K+1, ..., -1, 0, 1, 2, 3, ...$ We assume that the sequence $\{A_n\}$ is stationary, and independent of the sequence $\{C_n\}$. (We make no further assumptions on the sequence $\{C_n\}$). Finally, in order for the processes X_n and U_n to be well defined, we make assumptions on the initial conditions: we assume that

$$U_{-K}, ..., U_{-1} = 0, \text{ and } X_{-K} \text{ has some initial distribution } \gamma. \tag{5}$$

Let X^0 denote the queue length process corresponding to no delay in information, i.e. $K = 0$, and an initial distribution γ (the same as in (5)).

Proposition 2.1 *(i) We have* $X_n =_d X_n^0 + \sum_{i=1}^{K} A_{n-i+K} = X_n^0 + \sum_{i=n}^{n+K-1} A_i$.
(ii) If $\{A_n\}$ *are i.i.d. then* $X_n =_d X_n^0 + \sum_{i=1}^{K} \tilde{A}_i$, *where* $\tilde{A}_i$ *are i.i.d., distributed like* A_1, *and are independent of* X_n^0.

Remark 2.1 *Note that in (i),* X_n^0 *and* $\sum_{i=1}^{K} A_{n-i+K}$ *need not be independent. This follows from the fact that* X_n^0 *depends on the whole arrival process* $\{A_l, l < n\}$, *and since the sequence* $\{A_n\}$ *are not assumed to be i.i.d., the variables* $\{A_n, ..., A_{n+K-1}\}$ *also depend on* $\{A_l, l < n\}$, *and hence the possible dependence between* X_n^0 *and* $\sum_{i=0}^{K-1} A_{n+i}$.

As a consequence of Proposition 2.1, we may obtain explicit expressions for the PGF of the queue length in steady state, for the case where both the number of arrivals per slot and service durations are each an i.i.d. sequence, independent of each other. (This model is denoted in [10] as a Batch Arrival *Geo*/*G*/1 queue). Let τ_n be the number of slots required for serving the nth message. Denote the PGF of τ_1 and A_1 by $\tilde{\tau}(z) = E\left[z^{\tau_1}\right]$, $\tilde{A}(z) = E\left[z^{A_1}\right]$, $|z| \leq 1$. Let $\lambda = E[A_1]$, $\rho = \lambda E[\tau_1]$.

The queue length in a Batch Arrival *Geo*/*G*/1 discrete time queue with non-idling service (and no delay in information) evolves according to $X_{n+1}^0 = \max(X_n^0 - C_n, 0) + A_n$. X_n^0 is known to be stable if $\rho < 1$; in that case it converges in distribution to a finite random variable X^0, whose PGF is given by (see [10] p. 21)

$$\tilde{X}^0(z) = E\left[z^{X^0}\right] = \frac{(z-1)\tilde{\tau}(\tilde{A}(z))}{z - \tilde{\tau}(\tilde{A}(z))}(1-\rho), \ |z| \leq 1, \tag{6}$$

and

$$E\left[X^0\right] = \rho + \frac{E[\tau_1(\tau_1 - 1)]\lambda^2 + E[\tau_1]E[A_1(A_1 - 1)]}{2(1-\rho)}. \tag{7}$$

Combining (6), (7) with Proposition 2.1, we obtain

Corollary 2.1 *Assume that* $\rho < 1$, *and let* $\{A_n\}$ *and* $\{\tau_n\}$ *be i.i.d. and independent. Then the queue length for the system with* K *slots delayed information is stable: the queue length converges in distribution to a finite random variable* X^K, *whose PGF is given by*

$$\tilde{X}^K(z) = E\left[z^{X^K}\right] = \frac{(z-1)\tilde{\tau}(\tilde{A}(z))}{z - \tilde{\tau}(\tilde{A}(z))}(1-\rho)\,[\tilde{A}(z)]^K, \ |z| \leq 1, \tag{8}$$

$$E\left[X^K\right] = E\left[X^0\right] + K\lambda, \tag{9}$$

where $E\left[X^0\right]$ *is given in (7).*

3 Applying decomposition for vacation models

We need the following result (from [8]) on work decomposition for discrete time queues with batch arrivals and server vacations. A vacation is defined as a time period in which the server is not available for service. In Section 2, a vacation corresponds to a sequence of consecutive time-slots n for which $U_n = 0$, where U_n is defined in (2).

Consider the following discrete time Batch Arrival *Geo*/*G*/1 queue, with service vacations. During time slot n, a batch of A_n messages arrives. The number of packets in the mth arriving message (and hence the time duration in slots it takes to serve the mth message) is τ_m. $\{A_n\}$ and $\{\tau_m\}$ are i.i.d. sequences, independent of each other. Occasionally, the server takes a vacation.

We shall use the results from [8] for the more general vacation structure described in [9]. We make the following additional assumptions from [9]: (i) Each service time is independent of the sequence of vacation periods that preceded that service time. (ii) All customers (messages) arriving to the system are eventually served; the system has an infinite queueing capacity, and $\rho = E[A_1]E[\tau_1] < 1$. (iii) Customers are served in an order that is independent of their service times. (iv) Service is nonpreemptive. (v) The rules that govern when the server begins and ends vacations do not anticipate future jumps of the arrival process ("anticipate" in the sense of [13]), see Assumption 5 in [9].

Lemma 3.1 *In steady state, the amount of work W at the beginning of an arbitrary slot is equal in distribution to the sum of the following independent RVs: (i) V, the amount of work in the same queue at the beginning of an arbitrary slot when there are no vacations; (ii) Y, the amount of work at the beginning of an arbitrary non-serving slot. The same decomposition holds also for the queue length: $X =_d X_{nv} + Q$, where X_{nv} is the queue length in the beginning of an arbitrary slot in the queue without vacations and Q is the number of messages at the beginning of an arbitrary non-serving slot.*

We show how Corollary 2.1 can be obtained using the above decomposition. We first show that Q, the queue length at the beginning of an arbitrary non-serving slot, is distributed like $\sum_{i=1}^{K} A_i$. Assume that the server is busy till time $n-1$ and idles from time n (hence $U_n = 0$ and $U_{n-1}C_{n-1} = 1$) till time $m > n$ (at time slot m it restarts serving). It follows that all messages present at time $n-K$ have been served till the end of slot $n-1$, i.e. $X_{n-K} = \sum_{i=1}^{K} U_{n-i}C_{n-i}$. The number of messages in the queue at the beginning of the non-serving period, i.e. at time slot n, is thus $X_n = \sum_{i=n-K}^{n-1} A_i =_d \sum_{i=1}^{K} A_i$. Combining this with (6) and with Lemma 3.1, we obtain the result in Corollary 2.1.

Remark 3.1 In fact, one can show that at each idling slot, $n \le l \le m-1$, the number of messages in the queue is distributed like $\sum_{i=1}^{K} A_i$.

4 Delayed information on arrivals and service

In this section we study the case where information on both the queue size and messages' service completion is delayed $K = 1$ slots. Consequently, at time n, the value of C_{n-1} is unknown. The dynamics (1) stays unchanged, but the decision at time n whether to serve a message or not will differ from that of Section 2. We consider the following dynamics $U_n = 1(\{X_{n-1} > 1\} \cup \{X_{n-1} = 1, U_{n-1} = 0\})$. That is, the server attempts to serve the queue if either there were at least two messages at the beginning of time slot $n-1$, or if there was one message at that time but service was not attempted. Otherwise, the server takes a vacation of one slot, at the end of which new information is obtained.

We assume below that $\{A_n\}$ and $\{C_n\}$ are two independent i.i.d. sequences, and that C_n are Bernoulli with parameter μ (i.e. $C_n = 1$ w.p. $0 \le \mu \le 1$). We shall assume that the system operates in a steady state regime ($\rho < 1$).

We now compute the PGF of Q, the number of messages in steady state at an arbitrary non-serving period. Let n be the instant when a non-serving period begins. Hence, at the beginning of time slot $n-1$ there was still one message in the system, and service was attempted during time slot $n-1$. This means that at the beginning of time slot n, the queue contains the messages that have arrived during time slot $n-1$, plus (perhaps) the message whose service was attempted (and not necessarily completed) at time slot $n-1$. There are two possible types of non-serving periods, I_1 and I_2 (these will also be used to denote durations of these periods).

Type 1: If the service of the single message that was present in the system at time slot $n-1$ was completed at the end of that time slot, then at the beginning of time slot n, there are A_{n-1} messages present. The duration of the non-serving period is geometrically distributed with parameter $\tilde{A}(0)$ (the probability of no arrival in a time slot), and its expected duration is $1/(1-\tilde{A}(0))$.

Type 2: The service of the single message that was present in the system at time slot $n-1$ was not completed at the end of time slot $n-1$. At the beginning of time slot n, there are $A_{n-1}+1$ messages. The server takes a vacation of only one time slot, since at time slot $n+1$ the server will be aware that the message served at time $n-1$ is still in the system regardless of the value of A_{n-1}.

It follows that the expected duration of an arbitrary non-serving period I is

$$E[I] \;=\; E[I_1]P(C_1=1)+E[I_2]P(C_1=0)=\frac{1}{(1-\tilde{A}(0))}\times\mu+1\times(1-\mu).$$

The probability $P(I_2)$ that an arbitrary message that arrives during a non-serving period arrives to a type-2 non-serving period is $(1-\mu)/E[I]$, and the probability $P(I_1)$ that it arrives to a type-1 non-serving period is $1-(1-\mu)/E[I]$. Let $\tilde{Q}(z)$ denote the PGF of Q. Then

$$\begin{aligned}\tilde{Q}(z)=E\left[z^{Q}\right] &= E\left[z^{Q}\middle|I_1\right]P(I_1)+E\left[z^{Q}\middle|I_2\right]P(I_2) \qquad (10)\\ &= \tilde{A}(z)P(I_1)+\tilde{A}(z)zP(I_2)\\ &= \tilde{A}(z)\left(1+(z-1)(1-\mu)\left[\frac{\mu}{(1-\tilde{A}(0))}+1-\mu\right]^{-1}\right).\end{aligned}$$

Applying the decomposition from Lemma 3.1 and using (6), the PGF of the number of messages in the system at steady state is

$$\tilde{X}(z)=\frac{(z-1)\tilde{\tau}(\tilde{A}(z))}{z-\tilde{\tau}(\tilde{A}(z))}(1-\rho)\times\tilde{A}(z)\left(1+(z-1)(1-\mu)\left[\frac{\mu}{(1-\tilde{A}(0))}+1-\mu\right]^{-1}\right). \qquad (11)$$

The assumption that $\{C_n\}$ is a Bernouilli sequence is equivalent to assuming that $\{\tau_m\}$ is a sequence of geometric i.i.d. random variables, i.e., $P(\tau_m=j)=(1-\mu)^{j-1}\mu$, $j=1,2,3,\ldots$. The PGF, $\tilde{\tau}(z)$, of τ_1 and its moments are given by

$$\tilde{\tau}(z)=\frac{\mu z}{1-(1-\mu)z},\qquad E[\tau_1]=\mu^{-1},\qquad E[\tau_1(\tau_1-1)]=\frac{2(1-\mu)}{\mu^2}. \qquad (12)$$

The PGF of W, the amount of work in the system is then given by

$$E\left[z^{W}\right] \;=\; E\left\{E\left[z^{W}\middle|X\right]\right\}=E\left\{(E[z^{\tau_1}])^{X}\right\}=\tilde{X}\left(\tilde{\tau}(z)\right)=\tilde{X}\left(\frac{\mu z}{1-(1-\mu)z}\right).$$

5 Priority queues

Consider N queues and a single server. Time is slotted as before. Let $\{X_n = (X_n^1, ..., X_n^N)\}$, $n = -K, -K+1, ..., -1, 0, 1...$, be the process of queue lengths, where X_n^i denotes the number of messages in queue i at the beginning of time slot n. During time-slot n, a batch of A_n^i messages arrives to queue i. The arrivals to different queues at a time-slot may be correlated. U_n^i denotes whether service is attempted at time n at queue i. We consider the simpler case where all messages are of one unit length, so that if service is attempted at time n in queue i then a message leaves queue i at the end of that slot. This yields the following dynamics:

$$X_{n+1}^i = X_n^i + A_n^i - U_n^i, \qquad i = 1, ..., N \tag{13}$$

The service decisions are governed by the following rule:
(i) service is attempted at time n if the server is sure that the system is nonempty at that time. The available information for this decision is the queue lengths till time $n-K$, as well as the previous decisions $U_l = (U_l^1, ..., U_l^N)$ for $l < n$.
(ii) The queues have a (non-preemptive) priority order according to their indices, so that queue 1 has the highest priority and queue N has the lowest. This is reflected in the following evolution of U_n:

$$\begin{aligned} U_n^1 &= 1\{X_{n-K}^1 > \sum_{i=1}^K U_{n-i}^1\}, \\ U_n^2 &= 1\{U_n^1 = 0, X_{n-K}^2 > \sum_{i=1}^K U_{n-i}^2\}, \\ U_n^m &= 1\{U_n^l = 0, l = 1, ..., m-1, X_{n-K}^m > \sum_{i=1}^K U_{n-i}^m\}, \qquad m = 3, ..., N. \end{aligned} \tag{14}$$

We assume that $U_j^m = 0$ for $m = 1, ..., N$, $j = -K, -K+1, ..., -1$. The above enables one to express the distribution of the total number of messages belonging to queues with priority higher than or equal to l, $1 \le l \le N$. One can show by induction that, for every m, $1 \le m \le N$, and all $n \ge 0$, $X_{n-K}^m \ge \sum_{i=1}^K U_{n-i}^m$. Define now, for each l, $1 \le l \le N$, $X_n^*(l) = \sum_{i=1}^l X_n^i$, $A_n^*(l) = \sum_{i=1}^l A_n^i$, $U_n^*(l) = \sum_{i=1}^l U_n^i$. It follows that $X_{n-K}^*(l) > \sum_{i=1}^K U_{n-i}^*(l)$ if and only if, for some $r = 1, ..., l$, $X_{n-K}^r > \sum_{i=1}^K U_{n-i}^r$. It then follows that

$$X_{n+1}^*(l) = X_n^*(l) + A_n^*(l) - U_n^*(l), \qquad U_n^*(l) = 1\left\{X_{n-K}^*(l) > \sum_{i=1}^K U_{n-i}^*(l)\right\}. \tag{15}$$

(15) is a special case of the dynamics (1) and (2). It thus follows that we may apply Proposition 2.1 and Corollary 2.1 to obtain expressions for the distribution of $X_n^*(l)$ and its steady state version. Finally, in order to compute the expectation of X_n^m, the number of messages of priority m at time n, it suffices to note that $X_n^m = X_n^*(m) - X_n^*(m-1)$ and hence $E[X_n^m] = E[X_n^*(m)] - E[X_n^*(m-1)]$. A similar argument allows to compute

the expected number of messages of priority m in steady state (see (9)), the expressions for which are given in the sequel.

Optimal order of priority

Let $X^{K,m}$ be the steady state queue size of priority m when the delay is K slots, and let $X^{K,*}(m) = \sum_{l=1}^{m} X^{K,l}$, $X^{K,*}(0) = 0$. Then for $1 \leq m \leq N$, $X^{K,m} = X^{K,*}(m) - X^{K,*}(m-1)$. From (9),

$$E\left[X^{K,*}(m)\right] = E\left[X^{0,*}(m)\right] + K\left(\sum_{j=1}^{m} \lambda_j\right) \tag{16}$$

where $\lambda_j = E[A_1^j]$. Therefore,

$$E\left[X^{K,m}\right] = E\left[X^{0,m}\right] + K\lambda_m. \tag{17}$$

Let $W^{K,m}$ be the waiting time (until service) of class-m customer, where the delay is K slots. Then, since each message is of length one,

$$E\left[X^{0,m}\right] = \lambda_m\left(E\left[W^{0,m}\right] + 1\right) \quad \text{and} \quad E\left[W^{K,m}\right] = \frac{E\left[X^{K,m}\right]}{\lambda_m} - 1. \tag{18}$$

It follows that

$$\begin{aligned} E\left[W^{0,m}\right] \;=\; & \frac{\sum_{k=1}^{m} E[A_1^k(A_1^k - 1)]}{2\left(1 - \sum_{i=1}^{m-1}\lambda_i\right)\left(1 - \sum_{i=1}^{m}\lambda_i\right)} + \frac{\sum_{k=2}^{m}\sum_{j=1}^{k-1} E[A_1^k A_1^j]}{\left(1 - \sum_{i=1}^{m-1}\lambda_i\right)\left(1 - \sum_{i=1}^{m}\lambda_i\right)} \\ & + \frac{E[A_1^m(A_1^m - 1)] + 2\sum_{k=1}^{m-1} E[A_1^k A_1^m]}{2\lambda_m\left(1 - \sum_{i=1}^{m-1}\lambda_i\right)}, \end{aligned} \tag{19}$$

see [10] eq. (6.39), and [11, 12]. Combining (17), (18) and (19) yields explicit expressions for $E\left[X^{K,m}\right]$.

We now wish to order the priority classes such that the mean weighted waiting time of an arbitrary customer is minimized. That is, our objective is to

$$\text{Minimize}\left\{\sum_{m=1}^{N} c_m \frac{\lambda_m}{\lambda} E\left[W^{K,m}\right]\right\}, \tag{20}$$

where c_m is the rate of cost of class-m customers. By (17), (18),

$$E\left[W^{K,m}\right] = \frac{E\left[X^{0,m}\right]}{\lambda_m} + K - 1 = E\left[W^{0,m}\right] + K.$$

Hence, (20) is equivalent to minimizing $\sum_{m=1}^{N} c_m\lambda_m\left[E\left[W^{0,m}\right] + K\right]$. Since $\sum_{m=1}^{N} c_m\lambda_m + K$ is constant over all priority orders, the latter is minimized by the well known $c\mu$ rule which, in our case, assigns priorites by decreasing order of c_m. Hence, the queue with the highest c_m gets the highest priority, that is, the lowest index. This follows since the $c\mu$ rule is already optimal for the case of no-delay [6].

References

[1] E. Altman and G. Koole, "Control of a random walk with noisy delayed information", *Systems and Control Letters*, **24**, 207-213, 1995.

[2] E. Altman and G. Koole, "Stochastic scheduling games with Markov decision arrival processes", *J. Computers and Math. with Appl.*, 141-148, 1993.

[3] E. Altman and P. Nain, "Closed-Loop control with delayed information", *Performance Evaluation Review* **20**, 193-204, 1992.

[4] E. Altman and S. Stidham, "Optimality of monotonic policies for two-action Markovian decision processes", to appear in *QUESTA*, 1995.

[5] D. Artiges, "Optimal routing into two heterogeneous service stations with delayed information", to appear in IEEE-AC.

[6] J. S. Baras, D. -J. Ma, and A. M. Makowski, "K competing queues with geometric service requirements and linear costs: the μc rule is always optimal," *Systems and Control Letters* **6**, 173-180, 1985.

[7] O. J. Boxma, "Workloads and Waiting Times in Single-Server Systems with Multiple Customer Classes", *Queuing Systems*, **5**, 185-214, 1989.

[8] O. J. Boxma and W. P. Groenendijk, "Waiting times in discrete-time cyclic-server systems", *IEEE Transactions on Communications*, **36**, 164-170, 1988.

[9] S.W. Fuhrmann and R. R. Cooper, "Stochastic decomposition in the M/G/1 queue with generalized vacations", *Operations Research*, **33**, 1117-1129, 1985.

[10] H. Takagi, *Queueing Analysis, Vol. 3: Discrete-Time Systems*, North-Holland, 1993.

[11] Y. Takahashi and O. Hashida, "Delay analysis of discrete-time priority queue with structured inputs", *Queueing Systems*, **8**, 149-164, 1991.

[12] Y. Takahashi and O. Hashida, "Correction to our paper: Delay analysis of discrete-time priority queue with structured inputs", *QUESTA*, **13**, 449-450, 1993.

[13] R. W. Wolff, "Poisson arrivals see time averages", *Oper. Res.* **30**, 223-231, 1982.

DESIGNING A MULTICAST SWITCH SCHEDULER *

BALAJI PRABHAKAR NICK MCKEOWN †

ABSTRACT

This paper presents the design of the scheduler for an $M \times N$ input-queued switch. It is assumed that each input maintains a single queue for arriving multicast cells and that only the cell at the head of line (HOL) can be observed and scheduled at one time. The scheduler is required to be work-conserving, which means that no output port may be idle as long as there is an input cell destined to it. Furthermore, the scheduler is required to be fair, which means that no input cell may be held at HOL for more than M cell times (M is the number of input ports). The aim is to find a work-conserving, fair policy that delivers maximum throughput and minimizes input queue latency. When a scheduling policy decides which cells to schedule, contention may require that it leave a *residue* of cells to be scheduled in the next cell time. The selection of where to place the residue uniquely defines the scheduling policy. It is demonstrated that a policy which always concentrates the residue, subject to our fairness constraint, always outperforms all other policies. We present one such policy, called TATRA, and analyze it geometrically. We also present a heuristic round-robin policy called *mRRM* that is simple to implement in hardware, fair, and performs quite well when compared to a concentrating algorithm.

1. INTRODUCTION

A growing proportion of traffic on the Internet is multicast, with users distributing a wide variety of audio and video material. This dramatic change in the use of the Internet has been facilitated by the MBONE [1], [2], [3]. It seems inevitable that the volume of multicast traffic will continue to grow for sometime to come. So, if ATM switches are to find widespread use in the Internet, either as standalone switches, or as the core of high performance routers, it is important that they be able to handle multicast traffic efficiently. Although a number of different architectures and implementations have been proposed for multicast switches [6, 7, 8], we restrict our attention to input-queued switches. In particular, we consider how an input-queued switch may schedule multicast cells so as to achieve a high throughput and hence efficient utilization.

Most of the work on input-queued ATM switches has concentrated on unicast traffic in which cells are destined for only a single output. It is well known that when FIFO queues are used, the throughput of an input queued switch with unicast traffic can be limited to just 58% under relatively benign conditions [4]. When arrivals are correlated, the throughput can be even lower [5]. However, numerous papers have indicated that by using non-FIFO input queues and using good scheduling policies, much higher throughputs are possible [9, 10, 11, 12, 13, 14, 16, 17]. In [18], Hayes et

---

[*] Computer Systems Lab Technical Report, Stanford University.

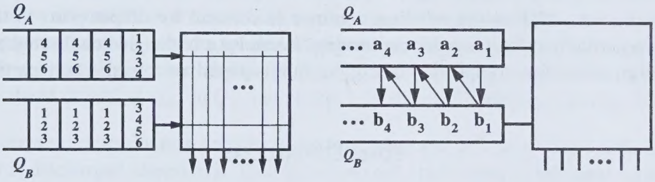
[†] Balaji Prabhakar is with the Basic Research Institute in the Mathematical Sciences (BRIMS), Hewlett-Packard Labs, Bristol. Email: balaji@lion.statslab.cam.ac.uk. Nick McKeown is with the Departments of Electrical Engineering and Computer Science, Stanford University. Email: nickm@ee.stanford.edu. This work was done when B. Prabhakar was a Visiting Scholar at the Department of EECS, UC Berkeley.

al. give an excellent queueing analysis of the performance of input-queued multicast switches. To maintain tractability, they assume a random scheduling policy for determining which cells are copied to each output during each cell time. Specifically, each output randomly and independently selects one input from among those that request it.

In this paper we consider the performance of different multicast scheduling policies. As may be expected, we find that the random scheduling policy is not the optimum policy. Instead, we find that a better algorithm is one that concentrates the cells that it leaves behind (the *residue*) on as few inputs as possible. In the next section we describe our model and various scheduling policies in more detail. We then prove that for a 2xN switch, the concentrating policy is the optimum policy. Following this, we present simulation results that demonstrate the optimality of the residue-concentrating policy. Finally, we describe TATRA - an efficient $M \times N$ multicast switch scheduler.

2. THE SWITCH MODEL

It is assumed that the switch has M input and N output ports and that each input maintains a single FIFO queue for arriving multicast cells. The input cells are assumed to contain a vector indicating which outputs the cell is to be sent to.



(a) (b)

Figure 1:

For example, the 2 input and N output switch shown in Figure 1a has a cell at the head of each queue. Input queue Q_A has an input cell destined for outputs $\{1, 2, 3, 4\}$ whereas input queue Q_B has an input cell destined for outputs $\{3, 4, 5, 6\}$. We shall refer to the size of the vector as the *fanout*. In the figure, the input cell at the head of each queue has a fanout of four. For clarity, we distinguish an arriving *input* cell from its corresponding *output* cells. In the figure, the input cell at the head of queue Q_A will generate four output cells.

The input queues are necessary because cells at different inputs may wish to copy cells to the same output port. At the end of each cell time, a scheduling policy decides which input cells to copy to which output ports. The policy selects a conflict-free match between input and output ports such that each output receives at most one cell. Thus, at the end of every cell time, the scheduling policy discharges some output cells, possibly leaving behind some residual output cells at the head-of-line (HOL) of the input buffers. For example, in the situation depicted in Figure 1a the *discharge* will consist of output cells for outputs $\{1, 2, 3, 4, 5, 6\}$, and the *residue* will consist of output cells for outputs $\{3, 4\}$. Therefore, the scheduling policy has to decide where to place the residue. It may elect to place the residue on both inputs (i.e., it "distributes the residue"), or it may place the residue on either Q_A only or on Q_B only (i.e., it "concentrates the residue"). It is the purpose of this paper to argue, based on theoretical results and simulations, that a scheduling policy that always "concentrates the residue" performs better (improves output utilization, reduces input queue latency, etc.) than one that does not always concentrate residue.

To reduce the implementation complexity, we assume that an input cell must wait in line until all of the cells ahead of it have gained access to all of the outputs that they requested. Furthermore, it is assumed that the scheduling policy observes only the cell at the head of each input queue, without further knowledge of the contents of individual input buffers behind the HOL and traffic arrival patterns. Perhaps the simplest way to service the input queues is to replicate the input cell over multiple cell times, generating one output cell per cell time. However, this service discipline does not take advantage of the multicast properties of the crossbar switch. So instead, we assume that one input cell can be copied to any number of outputs in a single cell time for which there is no conflict.

Following the description in [18], we distinguish two different service disciplines. The first is *full multicast* in which all of the copies of a cell must be sent in the same cell time. If any of the output cells loses the contention for an output port, none of the output cells are transmitted and the cell must try again in the next cell time. The second discipline is *partial multicast* in which case output cells may be delivered to output ports over any number of cell times. Only those output cells that are unsuccessful in one cell time continue to contend for output ports in the next cell time. Because partial multicast is work conserving, it enables a higher switch throughput, for little increase in implementation complexity. Therefore in this paper we consider only partial multicast policies.

3. SOME DEFINITIONS

Residue: The *residue* is the set of output cells that lose contention for output ports and remains at the HOL of the input queues at the end of each cell time. It is important to note that given a set of requests, every work-conserving policy will leave the same residue. However, it is up to the policy to determine how the residue is distributed over the inputs.

Concentrating Policy: A multicast scheduling policy is said to be *concentrating* if, at the end of every cell time, it leaves the residue on the smallest possible number of input ports.

Distributing Policy: A multicast scheduling policy is said to be *distributing* if, at the end of every cell time, it leaves the residue on the largest possible number of input ports.

A Non-concentrating Policy: A multicast scheduling policy is said to be *non-concentrating* if it does not always concentrate the residue.

Fairness Constraint: A multicast scheduling policy is said to be *fair* if each input cell is held at the HOL for no more than M cell times.

Note that in the two input case this definition of fairness means that the residue alternates between the two inputs. As will be seen, the fairness constraint leads to a round robin priority among the various inputs which is used to resolve contention between them. It can be seen that the above definition of fairness leads to a deterministic bound (of M cell times) on the latency of input cells at HOL.

4. MAIN RESULT

We state and briefly discuss the main results of the paper. The basic tenet of the paper is expressed below.

A work-conserving multicast switch scheduling policy that always "concentrates residue" at every possible instant subject to the fairness condition, leads to a higher output utilization than any other policy under arbitrary input processes.

Our goal is to demonstrate this observation and to use it to design an efficient multicast switch scheduler. Simulations are presented in Section 6 in support of this observation under various arrival process distributions; and a sample path proof is presented for the $2 \times N$ case. Specifically, we outline a proof of the following theorem.

Theorem 1: *A scheduling policy for a $2 \times N$ multicast switch that always "concentrates residue" at every possible instant subject to a natural fairness condition, performs better than any other policy.*

Outline of proof: An outline of the proof of Theorem 1 is given under the static input assumption. That is, it will be assumed that at time 0 both input queues have an infinite number of packets placed according to some (possibly random) packet configuration. Fix one such configuration and suppose that the cells in inputs 1 and 2 are labelled $\{a_i\}_{i=1,2,...}$ and $\{b_i\}_{i=1,2,...}$ respectively; where, for all i, packet a_i is ahead of packet a_{i+1} in Q_A and packet b_i is ahead of packet b_{i+1} in Q_B. See Figure 1b.

The fairness constraint now reduces to the following: the cell (or residue) at the HOL of each input buffer is discharged alternately. That is, no two cells belonging to the same input can depart in successive cell times without there being an input cell departure from the other queue. This leads to the following ordering of input cells: (1) $a_1 \leq_d b_1 \leq_d a_2 \leq_d b_2 \ldots$, if a_1 is the first cell to depart; and (2) $b_1 \leq_d a_1 \leq_d b_2 \leq_d a_2 \ldots$, if b_1 is the first cell to depart.

Without loss of generality, we will adopt the first contention resolution scheme and *link* input cells in a vertical or oblique fashion as shown in Figure 1b. The directions of the arrows on the links denote where the residue is to be concentrated, should a policy choose to concentrate residue at some time. We are only interested in fair scheduling policies as defined above, and so restrict our attention to such policies in this paper. Let the vertical link between a_i and b_i be labelled l_{2i-1} and the oblique link between b_i and a_{i+1} be labelled l_{2i}. The following facts now follow easily.

Fact 1: *All scheduling policies work their way through links $l_1, l_2, l_3, \ldots$ in that order releasing no links (when there is contention between cells at HOL and residue is distributed), one link (when there is contention between cells at HOL and residue is concentrated), or two links (when there is no contention between cells at HOL) in one cell time.*

Fact 2: *The time at which an input cell is completely served is exactly equal to the time at which the link emanating from it is released.*

To prove the theorem, it is sufficient to show that a fair concentrating policy π^* releases each link i before any other fair policy π. The details of how this is accomplished may be found in [15].

Discussion: We now offer an intuitive explanation as to why a policy that always concentrates the residue outperforms one that does not. Referring to Figure 1a, consider the options faced by a work-conserving scheduling algorithm at this time (t_1). Note that whatever decision the algorithm makes, the residue will be the same. The scheduling algorithm just determines where to place the residue. If at time t_1, the algorithm concentrates the residue on Q_B then all of a_1's output cells will be sent and cell a_2 will be brought forward at time t_2. At time t_2, the algorithm selects between a_2 *and* the residue leftover from t_1. If on the other hand, the algorithm distributes the residue over both input queues at t_1, then at t_2 the algorithm can only schedule the residue leftover from t_1. No new cells can be brought forward. So, *on average*, a concentrating policy will bring new work forward sooner, increasing the diversity of its choice. This enables it to schedule more output cells in the next cell time.

To demonstrate that the fairness constraint is necessary, consider the example of Figure 1a again. Assume that the concentrating policy is *not* fair and concentrates the residue at Q_A at both times t_1 and t_2. From then on, only one input cell can be completed per cell time. An algorithm that distributes residue at time t_1 would actually perform better. However, if the concentrating policy is fair and at time t_2 concentrates the residue at Q_B, it will, in this case, complete input cells at the same time as the distributing policy. Thus, in addition to being socially correct, the fairness constraint provides a means for comparing scheduling policies pathwise, *instantaneously*.

5. SIMULATION RESULTS

The following simulations support the observation that a "concentrating policy" outperforms other fair policies.

5.1. SCHEDULING POLICIES

The four scheduling policies that we compare are as follows:

CONCENTRATE: This policy always concentrates the residue onto as few inputs as possible. This is achieved by performing the following algorithm at the beginning of each cell time.

1. Determine the residue.
2. Find the input with the most in common with the residue. If there is a choice of inputs, select the one with the input cell that has been at the HOL for the shortest time. This ensures some fairness though not in the sense of the definition in Section 3 (see remark below).
3. Concentrate as much residue onto the input as possible.
4. Remove the input from further consideration.
5. Repeat steps (2)-(4) until no residue remains.

Remark: The allocation of residue uniquely defines which input queues are served. This algorithm does not guarantee that each input cell will remain at HOL for no more M cell times. In Section 6 we outline an algorithm, TATRA, which offers this guarantee in addition to concentrating residue in an efficient manner.

DISTRIBUTE: This policy always distributes the residue onto as many inputs as possible. This is achieved by the following algorithm.

1. Determine the residue.

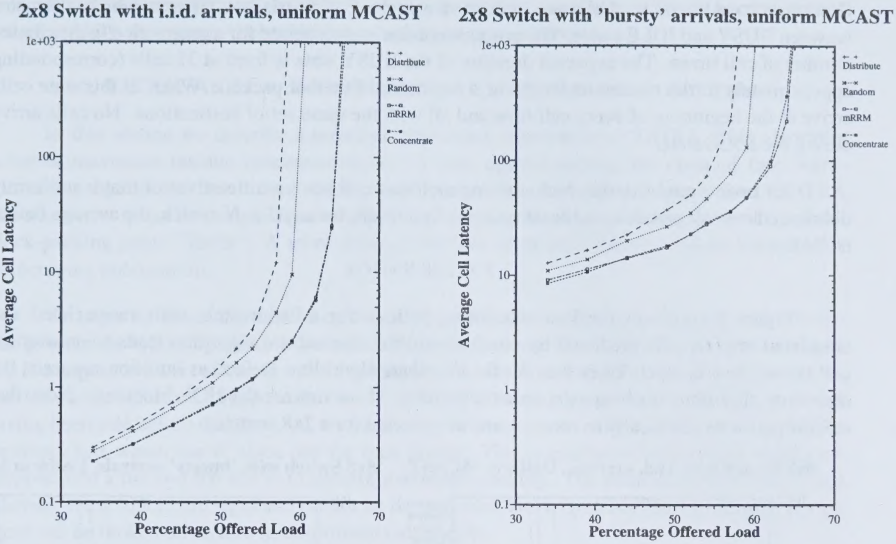


Figure 2: *Graph of average cell latency as a function of offered load for a 2x8 switch. Uncorrelated arrivals (left). Correlated arrivals (right).*

2. Find the input with at least one cell but otherwise the least in common with the residue. If there is a choice of inputs, select the one with the input cell that has been at the HOL for the shortest time.
3. Place one output cell of residue onto that input.
4. Remove the input from further consideration.
5. Repeat steps (2)-(4) until no inputs remain.
6. If residue remains, consider all the inputs again and start at step (2).

RANDOM: This policy is motivated by the work of Hayes et al. in [18], which is the multicast version of the algorithms described in [4] and [9]. Each output in turn, and independently of the other outputs, randomly selects one input from among those that request it.

MULTICAST ROUND ROBIN (*mRRM*): This policy is motivated by the algorithms described in [10, 11]. A single round-robin pointer is collectively maintained by all of the outputs. Each output selects the next input that requests it at, or after, the pointer. At the end of the cell time, the pointer is moved to one position beyond the first input that is served. Designed to be simple to implement in hardware, *mRRM* tends to concentrate the selection onto a small number of inputs, yet maintain fairness. Note that for a 2xN switch this algorithm performs almost identically to the *concentrate* algorithm.

5.2. TRAFFIC TYPES

We compare each scheduling policy for two different arrival processes:

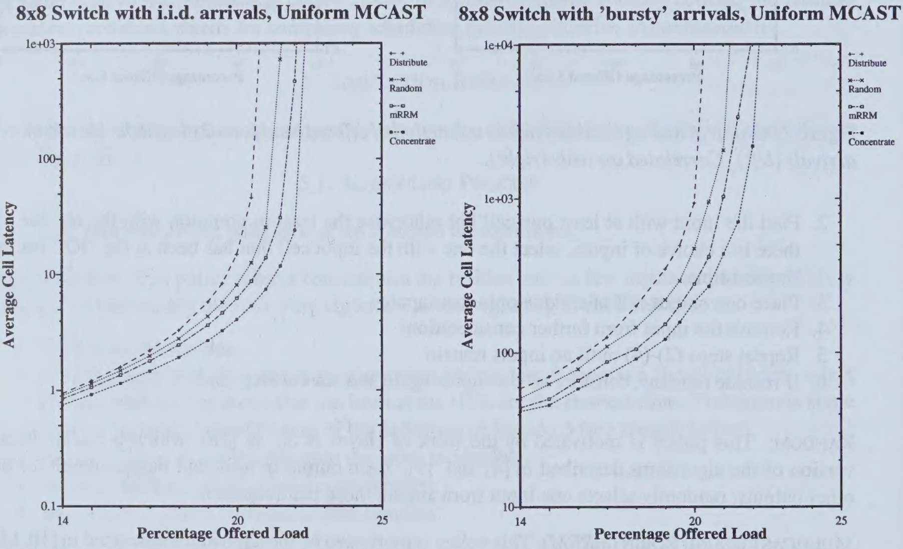
UNCORRELATED ARRIVALS: At the beginning of each cell time, a cell arrives at each input with probability p independently of whether a cell arrived during the previous cell time.

CORRELATED ARRIVALS: Cells are generated using a 2-state Markov process which alternates between BUSY and IDLE states. The process remains in each period for a geometrically distributed number of cell times. The expected duration of the BUSY state is fixed at 32 cells (corresponding approximately to the maximum length of a segmented Ethernet packet). When in this state cells arrive at the beginning of every cell time and all with the same set of destinations. No cells arrive during the IDLE state.

For both types of traffic, each arriving multicast cell has a multicast vector that is uniformly distributed over all possible multicast vectors. As a result, for an $M \times N$ switch, the average fanout is $N/2$.

5.3. 2X8 SWITCH

Figure 2 compares the four scheduling policies for a 2x8 switch, with uncorrelated and correlated arrivals. As predicted by our theorem, the *concentrate* algorithm leads to an average cell latency that is much lower than for the *distribute* algorithm. In fact, as intuition suggests, the *distribute* algorithm is always the worst algorithm: it maximizes the HOL blocking. Note that *mRRM* performs identically to *concentrate*, as expected for a 2x8 switch.


8x8 Switch with i.i.d. arrivals, Uniform MCAST

8x8 Switch with 'bursty' arrivals, Uniform MCAST

Figure 3: *Graph of average cell latency as a function of offered load for an 8x8 switch. Uncorrelated arrivals (left). Correlated arrivals (right).*

5.4. 8X8 SWITCH

Figure 3 compares the four scheduling policies for an 8x8 switch, with uncorrelated and correlated arrivals. Once again, the *concentrate* algorithm leads to an average cell latency that is much lower than for the *distribute* algorithm. This supports our argument, not proved in this paper, that the *concentrate* policy outperforms other algorithms.

Note that for an 8x8 switch *mRRM* performs worse than *concentrate*. This is because it does not necessarily concentrate the residue on as small a number of inputs.

6. TATRA: AN $M \times N$ MULTICAST SWITCH SCHEDULING ALGORITHM

In this section we describe a multicast scheduling algorithms — TATRA. This algorithm achieves maximum residue concentration and is thus optimal among the class of fair, work-conserving algorithms. Due to limitations of space, we only provide a brief description of TATRA and show how it can be analyzed geometrically by mapping it onto a game similar to the popular block-packing game "Tetris". A more detailed analysis of its performance will be presented in forthcoming publications.

We begin by observing that the operation of a multicast switch can be mapped into a Tetris-like game in the following sense. Input cells will be mapped into Tetris blocks and since each input cell is composed of a set of output cells, this Tetris block will be an amalgamation of smaller blocks, one for each output cell. Suppose that an $M \times N$ switch begins processing input cells at time 1 having been idle before that time. At the beginning of time 1, input cells at HOL are dropped into an empty box which has N slots, one for each output. This is similar to Tetris where blocks are dropped into a bin and the aim is to achieve maximum packing. The main difference here is that whereas Tetris blocks are rigid and cannot be decomposed, work conservation will require that an input cell be broken down into its constituent output cells.

We now explain what work conservation and fairness mean in this context. If an input has a cell destined to output j, then this output cell is dropped into the slot corresponding to the j^{th} output. The output cell will drop to the lowest unoccupied position in that slot. This ensures *work conservation,* since gaps in the output slots lead to an idling of outputs. The order in which input cells are dropped is given by the priority rule arising from the *fairness constraint.* All scheduling policies must obey these rules. Figure 4a traces the evolution of a 5×5 switch operating under a sub-optimal policy π and Figure 4b traces its evolution under the optimal policy TATRA.

Evolution of the switch under π: At the beginning of time 1 input 1 drops cells for outputs 1, 3 and 5 into the corresponding output slots. After input 1, input 2 drops its output cells into slots 2, 3, 4 and 5. (Without loss of generality, we have assumed that input i has priority over input $i + 1$). Proceeding thus, all 5 input cells are dropped into the box.

At the end of time 1, all output cells at the bottom-most layer of the box are discharged. That is, they are assumed to be served. Again, for the example in Figure 4a, this means that input 1 is completely served and can advance a new cell to HOL at time 2. Input 2 manages to discharge cells to outputs 2 and 4 and is left with a residue for outputs 3 and 5. Note that the *discharge* at any time is the set of output cells in the bottom-most layer and the *residue* is everything that's left behind.

At the beginning of time 2, all residue cells drop down one level. The new cell at input 1 drops into the box taking its place behind the residue in the manner described above. In Figure 4 the shaded cells are new input cells. Under π, at the end of time 2, input cell 2 is completely discharged along with a part of input cell 3. This goes on until *in four cell times π discharges all five original input cells and at least one fresh input cell (from input 1).*

Evolution of the switch under TATRA: During the first time instant, TATRA advances input cell 5 all the way to the bottom giving it priority over cells from inputs 2, 3 and 4; thus allowing it to depart at the end of time 1. Note that this does not conflict with the priority rule since it is impossible for inputs 2, 3 and 4 to leave at the end of time 1.

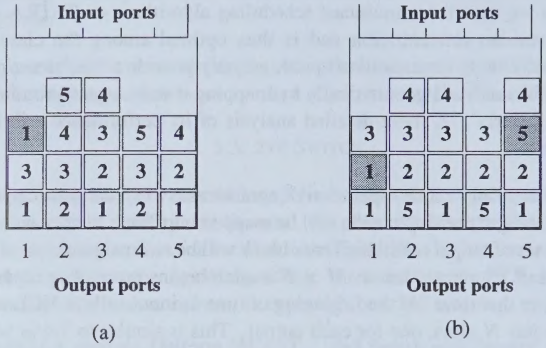


Figure 4:

At the beginning of time 2, under TATRA, two new cells (from inputs 1 and 5) are brought forth. Suppose that the order in which they drop down is again given by the priority rule (that is, 1 before 5)[1] Again, without delaying the departure of the original cells from inputs 2, 3 and 4, the priority of new cells from inputs 1 and 5 can be bumped up as shown in Figure 4b.

Thus, *in four cell times TATRA discharges all five original input cells and at least two new cells (from inputs 1 and 5). Furthermore, individual input cells are served earlier under TATRA than under π.* The following observation is the key aspect of multicast switching (subject to fairness, work conservation and knowledge of HOL cells only) exploited by TATRA in scheduling output cells:

No input cell is completely served until every output cell that belongs to it has been discharged. This means that no input cell is completely served until the output cell belonging to it that is furthest from the bottom of the box has departed. Call this furthest output cell the *peak cell.* (In Figure 4 cells from input 1, 2, 3 and 4 destined to output 3 are peak cells.) Therefore, it pays to bump the priority of a fresh input cell j over that of an input cell i which is already in the box *so long as the peak cell of i is not raised in the process.*

7. CONCLUSION

Scheduling policies for input-queued multicast switches have been studied. We observed that when designing a multicast scheduling policy, it is important to determine the placement of the residue. Subject to a natural fairness constraint, it is shown that for a $2 \times N$ switch the optimum policy is one that always concentrates the residue. Our simulation results indicate that the concentrating policy also outperforms a distributing or random policy for $M \times N$ switches. In addition, we presented two fair algorithms: (1) *mRRM* which performs favorably when compared to the concentrating algorithm, yet is simple to implement in hardware; and (2) TATRA which achieves maximum residue-concentration.

---

[1] It turns out that this ordering does not matter, since there are never more than M input cells in the box.

References

[1] Paxson, V; "Growth trends in wide-area TCP connections," *IEEE Network*, vol.8, (no.4):8-17. July-Aug 1994.

[2] Eriksson, H.; "MBone: the Multicast Backbone," *Communications of the ACM*, vol.37, (no.8):54-60. Aug 1994.

[3] Deering, S.E.; Cheriton, D.R.; "Multicast Routing in datagram internetworks and extended LANs," *ACM Transactions on Computer Systems*, vol.8, (no.2):85-110. May 1990.

[4] Karol, M., Hluchyj, M., and Morgan, S. "Input Versus Output Queueing on a Space Division Switch," *IEEE Trans. Comm*, 35(12) pp.1347-1356

[5] Li, S.-Q; "Performance of a nonblocking space-division packet switch with correlated input traffic," *IEEE Trans. Comm*, vol.40, (no.1):97-108. Jan 1992.

[6] Lee, T.T.; "Nonblocking copy networks for multicast packet switching," *IEEE J. Select. Areas Comm.*, vol.6, pp.1455-1467. Dec 1988.

[7] Turner, J.S.; "Design of a broadcast switching network," *Proc. IEEE INFOCOM '86*, pp.667-675.

[8] Huang, A.; "Starlite: A wideband digital switch," *Proc. IEEE GLOBECOM '84*, pp.121-125.

[9] Anderson, T., Owicki, S., Saxe, J., and Thacker, C. "High Speed Switch Scheduling for Local Area Networks," *Proc. Fifth International Conference on Architectural Support for Programming Languages and Operating Systems* Oct 1992, pp. 98–110.

[10] McKeown, N.; Varaiya, P.; and Walrand, J.; "Scheduling Cells in an Input-Queued Switch," *IEE Electronics Letters*, Dec 9th 1993, pp.2174-5.

[11] McKeown, N.; "Scheduling Algorithms for Input-Queued Cell Switches," PhD Thesis, University of California at Berkeley, May 1995.

[12] Chen, M.; Georganas, N.D.; "A Fast Algorithm for multi-channel/port traffic scheduling," *Proc. IEEE Supercomm/ICC '94*, pp.96-100.

[13] Obara, H. "An Efficient Contention Resolution Algorithm for Input Queueing ATM Switches," *Intl. Jour. of Digital & Analog Cabled Systems*, vol. 2, no. 4, Oct-Dec 1989, pp. 261-267.

[14] Obara, H. "Optimum Architecture For Input Queueing ATM Switches," *Elect. Letters*, 28th March 1991, pp.555-557.

[15] McKeown, N. and Prabhakar, B. "Scheduling Multicast Cells in an Input-Queued Switch," *Technical Report: Computer Systems Lab, Stanford University.*

[16] Obara, H., Okamoto, S., and Hamazumi, Y. "Input and Output Queueing ATM Switch Architecture with Spatial and Temporal Slot Reservation Control" *Elect. Letters*, 2nd Jan 1992, pp.22-24.

[17] Karol, M., Eng, K., Obara, H. "Improving the Performance of Input-Queued ATM Packet Switches," *INFOCOM '92*, pp.110-115.

[18] Hayes, J.F; Breault, R.; and Mehmet-Ali, M; "Performance Analysis of a Multicast Switch," *IEEE Trans. Commun., vol.39, no.4*, pp. 581-587. April 1991.

EM Algorithm and Recurrent Networks

Sheng Ma and Chuanyi Ji
Department of Electrical, Computer and System Engineering
Rensselaer Polytechnic Institute, Troy, NY 12180

Abstract

A stochastic model is established for fully-connected recurrent neural networks with sigmoid units based on Gibbs distributions. EM (Expectation-Maximization) algorithm is then applied to train recurrent networks through hidden state estimation. Using a mean-field approximation, training the original fully-connected recurrent network is reduced to training an array of individual feedforward neurons, which can be done easily by using fast training algorithms. The resulting EM-based algorithm significantly simplifies the original training process and reduces the training time.

1 Introduction

Fully connected recurrent neural networks are a class of neural networks with arbitrarily interconnected (nonlinear) sigmoid neurons. Similar to feedforward networks which are universal approximators for static functions, recurrent networks have been shown to be universal approximators for dynamical systems[11]. However, different from feedforward networks for which numerous probabilistic models have been developed based on information theory and statistical methods[7], similar results are missing for recurrent networks with sigmoid units. Therefore, most of the existing training algorithms for recurrent networks are derived in an ad hoc way. These algorithms are not very efficient due to complicated nature of recurrent networks.

The goal of this work is two-fold. First, we would like to develop a stochastic model to train recurrent networks with sigmoid units. Second, through the model developed, we will derive a novel training algorithm for recurrent networks which drastically simplifies the original training process.

2 A Stochastic Model

Consider a recurrent network with d inputs, L hidden units and one output unit (see figure 1). Let $x(n) \in R^d$ and $t(n+1) \in R^1$ be an input and a desired output of a recurrent network at n-th (and $n+1$-th) epoch, respectively. They are random variables drawn from some (underlying) stochastic processes. Let $z(n+1) \in R^L$ be a desired output of hidden units at $n+1$-th epoch, which are also called desired hidden states when the recurrent hidden units are regarded as state neurons. Let $\{x\}$, $\{z\}$ and $\{t\}$ denote all the inputs, desired hidden states and outputs up to epoch N. A stochastic model can then be established through a conditional probability model based on the Markov property of recurrent networks [1]

$$P(\{z\} \mid \{t\}, \{x\}, z(1), \Theta) = \prod_{n=1}^{N} P(z(n+1) \mid z(n), t(n+1), x(n), \Theta) \quad (1)$$

$$P(\{z\}, \{t\} \mid \{x\}, z(1), \Theta) = \prod_{n=1}^{N} P(z(n+1), t(n+1) \mid z(n), x(n), \Theta), \quad (2)$$

where Θ contains all the parameters of the recurrent network: $w^{(1)}$, $w^{(3)}$ and $w^{(2)}$, the weights between inputs and hidden units, between hidden units, and between hidden and output units, respectively. $z(1)$ is the initial desired hidden states. Furthermore, Gibbs distributions can be used which eventually lead to the following probabilities

$$P(z(n+1) \mid z(n), x(n), \Theta) = A \exp(-\frac{1}{2}(z(n+1) - h(n+1))^T \Sigma^{-1}(z(n+1) - h(n+1))), \quad (3)$$

$$P(z(n+1), t(n+1) \mid z(n), x(n), \Theta) = B \exp(-\lambda_1 E_1(n+1) - \lambda_2 E_2(n+1)), \quad (4)$$

where $E_1(n+1) = \| z(n+1) - h(n+1) \|^2$ and $E_2(n+1) = (t(n+1) - z(n+1)^T w^{(2)})^2$. λ_1, λ_2, A and B are constants. $h(n+1)$ is the actual output of sigmoid hidden units at $n+1$-th epoch. Then putting these four equations together, the two conditional probabilities are obtained as products of conditionally independent joint Gaussian distributions.

3 EM Algorithm For Recurrent Networks

Once the stochastic model is developed, a new training algorithm is derived through EM algorithm[4]. The essence of EM algorithm is that certain hidden variables (missing data) can be introduced to simplify a maximum likelihood problem. For our case, the hidden variables are desired hidden states $z(n+1)$'s which serve as missing data, whereas the incomplete data consists of pairs of $\{x(n), t(n+1)\}$'s. Using similar derivations as in [6], we can obtain the expected *log* likelihood

$$Q(\Theta \mid \Theta^p) = \int_{\{z\}} P(\{z\} \mid \{t\}, \{x\}, \Theta^p) \ln P(\{z\} \mid \{t\}, \{x\}, \Theta), \quad (5)$$

where Θ and Θ^p are the new parameters and the parameters at the previous step, respectively. Since $Q(\Theta \mid \Theta^p)$ is difficult to evaluate directly, a mean-field approximation[12] is used to estimate $Q(\Theta \mid \Theta^p)$, which eventually leads to an EM-based algorithm for recurrent networks.

E-step: Evaluate the expected desired hidden states recurssively through

$$\mathbf{E}z_j(n+1) = \hat{h}_j(n+1) + ae(n+1), \quad (6)$$

where $\hat{h}_j(n+1) = g(x(n)^T w_j^{(1)} + \mathbf{E}z(n)^T w_j^{(3)})$ for $1 \le j \le L$. $g(u)$ is a sigmoid function. $e(n+1) = t(n+1) - \hat{h}(n+1)^T w^{(2)}$, and a is a constant.

M-step: Using the expected desired hidden states obtained at the E-step as targets for recurrent hidden neurons to find new parameters through two steps.

(a) Find $w_j^{(1)}$'s and $w_j^{(3)}$'s through minimizing the difference between expected desired and "actual" hidden states $\hat{h}(n)$: $\sum_n \| \mathbf{E}z(n) - \hat{h}(n) \|^2$.

---

[1] This property comes from the fact that the outputs of a recurrent network and its hidden units at current epoch only depend on the actual outputs of hidden units at previous epoch.

(b) Find $w^{(2)}$ through minimizing difference between desired outputs of the network and weighted expected desired hidden targets: $\sum_n \left(\mathbf{E}z(n+1)^T w^{(2)} - t(n+1)\right)^2$.

The algorithm will iterate between the E- and M-steps until a convergence criterion is achieved.

Notice that (*a*) and (*b*) are equivalent to training individual feedforward neurons and can be solved using a fast training algorithm given in [3].

4 A Fast Algorithm to Train a Single Neuron

Since the major computational cost of our algorithm comes from M-step, i.e. training individual neurons, we need a fast training algorithm to train a single neuron. Several methods have been developed such as Breiman et. al. [3] and Shadafan et. al. [10]. However, the former can only handle ramp active function and the latter has complicate heuristic criterions to assist learning. Therefore, a fast training algorithm which is particularly suitable to our case is developed as follow.

Let $\{\vec{\tilde{x}}(n), \tilde{t}(n)\}_{n=1}^N$ be the training data to train a single neuron, where $\vec{\tilde{x}}(n)$ and $\tilde{t}(n)$ are the nth input vector and its corresponding target of this neuron respectively. The training task is to find an optimum $\vec{w}_{opt}$, such that

$$\vec{w}_{opt} = \text{argmin}_{\{\vec{w}\}} \sum_{n=1}^{N} (\tilde{t}(n) - g(\vec{w} \cdot \vec{\tilde{x}}(n)))^2, \tag{7}$$

where $\vec{w}$ is a weight vector of this neuron.

Assuming that $g^{-1}(\tilde{t}(n))$ is close to $\vec{w} \cdot \vec{\tilde{x}}(n)$, we can expend $g(\vec{w} \cdot \vec{\tilde{x}}(n))$ in Taylor series. By ignoring the second and higher order terms, we have:

$$\vec{w}_{opt} \approx \text{argmin}_{\{\vec{w}\}} \sum_{n=1}^{N} g'(g^{-1}\tilde{t}(n)))(\vec{w} \cdot \vec{\tilde{x}}(n) - g^{-1}(\tilde{t}(n)))^2. \tag{8}$$

Therefore, the nonlinear optimization problem (Eq.7) is reduced to the weighted linear regression problem (Eq.8) which has a closed-form solution. In practice, the assumption needed above can be made true through controlling the value of a.

5 Relations To The Other Works

Most of the related work for dynamical systems consider discrete states. For instance, statistical mechanics and Gibbs distributions were used in Boltzmann machines with discrete stochastic units[1]. Hidden Markov Models were considered in Baum-Welch algorithm for unsupervised learning for discrete states[8]. EM algorithm was applied to supervised learning for a (static) gating network with sigmoid units by Jordan et.al. [6]. In an independent work by Bengio et.al.[2], EM algorithm was used in a somewhat similar fashion to HMM with discrete states for supervised learning. The work given in this paper is for recurrent neural networks with continuous state neurons in an supervised learning environment. When GEM (Generalized EM) is appled in M-step, our algorithm is essentially similar to the moving target algorithm for recurrent networks by Rohwer[9].

	BPTT	EMR
4-hidden unit target network	0.000502[1] (0.000148[2]) 8.22min [3]	0.000831(0.000408) 0.771min
8-hidden unit target network	0.001023(0.000425) 16.21min	0.000954(0.000597) 1.21min
System Identification Problem	1.076(0.048) 178min	0.984(0.252) 16.3min

Table 1: 1: The Mean Square Error on the test set. 2: The Standard Deviation 3: The CPU-time on SPARC 10. All results are averaged over twenty runs.

6 Simulation Results

In order to test our algorithm and compare its performance with that of other methods, we choose two test problems: (1) learning underlying target recurrent networks; and (2) a nonlinear system identification problem used by Horne and Giles [5]. In the problem of learning underlying target recurrent networks, the weights of target networks are randomly chosen. Then input sequences are randomly generated from a uniform distribution over the interval $[-8.0, 8.0]$. Their corresponding outputs from a target network are used as the desired outputs in a training set. The advantage of using such underlying recurrent networks is that we can control the complexity of the problem easily by changing the number of recurrent hidden units of an underlying target network. Specifically, target recurrent networks with 4 and 8 fully connected hidden units are used in our experiments with one sequence of length 400 for training and 800 for testing. The results are given in Figure 2*a* (for 4-hidden-node target network) and 2*b* (for 8-hidden-node target network) as well as Table 1.

The second test problem, a nonlinear system identification problem, can be described as follows. Let $s_1(k)$ and $s_2(k)$ be two states of the system at time k. Let $u(k)$ and $t(k)$ be an input and the corresponding output of the system at time k ($k \geq 0$). The underlying mapping between input $u(k)$ and output $t(k)$ is then given by

$$\begin{aligned} s_1(k+1) &= \frac{s_1(k) + 2s_2(k)}{1 + s_2^2(k)} + u(k), \\ s_2(k+1) &= \frac{s_1(k)s_2(k)}{1 + s_2^2(k)} + u(k), \\ t(k+1) &= s_1(k+1) + s_2(k+1). \end{aligned}$$

To train recurrent networks to learn such a mapping, 1000 random training sequences of length 50 are generated as training samples, and the input sequences, $u(k)$'s, are randomly selected from $[-2., 2.]$ and the corresponding $t(k)$'s are used as the desired targets. Fully connected recurrent networks with 6 hidden neurons are used. The trained network is tested on a sine wave of frequency 0.04 rad/second which is totally different from the training sequences. The results are also given in Figure 2*c* as well as in Table 1.

The performance [4] of our algorithm is compared with that of BPTT (Back-Propagation Through

---

[2] The Mean Squared Error on Testing set.

[3] The cup-time. The algorithm is considered to be stop if no further improvements.

[4] All the results are average over 5 random runs on SPARC 10.

Time). The results show that the performance of EM algorithm is comparable with that of the other algorithms. The Training time, however, has been improved by at least 15 times.

7 Future Works

Our on-going efforts are directed towards exploring how to further investigate properties of recurrent neural networks using stochastic processes and how to further improve both performance and speed of our EM-based algorithm on more complicated problems.

Acknowledgement

The authors would like to thank Tal Grossman, Baral Pearkmutter and James Modestino for suggesting the related references and many helpful discussions. The support from National Science Foundation (ECS-9312504) is gratefully acknowledged.

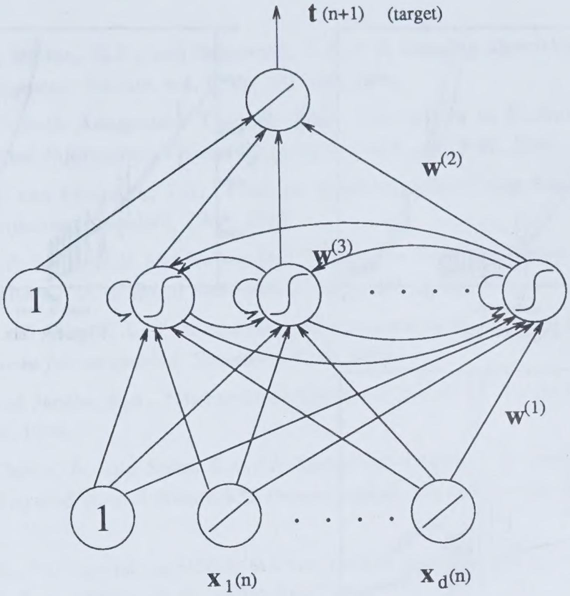

t (n+1) (target)
W(2)
W(3)
1
W(1)
1
x 1(n)
x d(n)

Figure 1: The Structure of a Fully Connected Recurrent Network

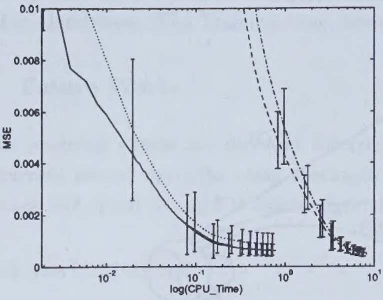


Figure: 2a

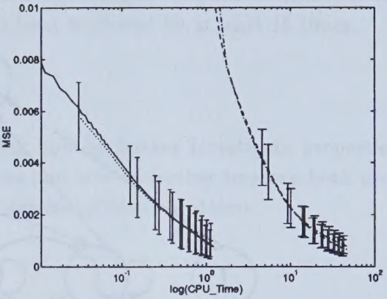


Figure: 2b

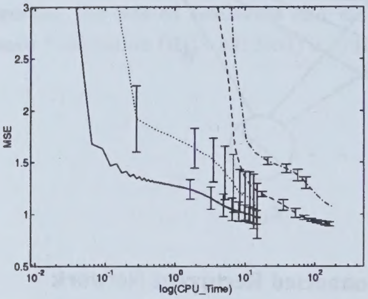


Figure: 2c

MSE: averaged Mean Square Error over 20 random runs. log(CPU_Time): the log CPU-Time in minute on *SPARC* 10 sun workstation. “-.” , “–”, “..” and “-” represent the training and testing error of BPTT and EMR respectively. The error bar, “I”, represents the standard derivation.

References

[1] Ackley, D.H., Hinton, G.E., and Sejnowski, T.J., " A learning algorithm for Boltzmann machines", *Cognitive Science*, vol. 9, pp. 147-169, 1985.

[2] Bengio, Y., "Credit Assignment Through Time: Alternative to Backpropagation," *Advances in Neural Information Processing Systems*, vol.6, pp 75-82, 1994.

[3] Breiman, L.E and Friedman, J.H, "Function Approximation Using Ramps" *Neural Networks for computing, Snowbird, Utah*, 1993.

[4] Dempster, A.P, Laird, N.M and Rubin, D.B "Maximum Likelihood from Incomplete Data via EM Algorithm," *J. of Royal Statistical Society*, B39, 1-33,1977.

[5] Horne, B.G. and Giles, C.L., "An experimental comparison of recurrent neural networks", *Neural Networks for computing, Snowbird, Utah*, 1993.

[6] Jordan, M. and Jacobs, R.A., "Hierarchical Mixture of Experts," *Neural Computation*, vol. 6, pp 181-214, 1994.

[7] Levin, E., Tishby, N. and Solla, S.A.,"A Statistical Approach to Learning and Generalization in Layered Neural Network", *Proceedings of the IEEE*, v78, No. 10 1568-1574, 1990.

[8] Rabiner, L.R., "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, pp257-285 , 1989.

[9] Rohwer, R., "The Moving Target Training Algorithm", *Advances in Neural Information Processing Systems 2*, pp 558-565, 1990.

[10] Shadafan, R.S. and Niranjan, M. "A Dynamic Neural Network Architecture by Sequential Partitioning of the Input Space", *Neural Computation* 6, 1203-1222, 1994.

[11] Sontag, E., "Systems combining linearity and saturations and relations to neural networks,"Tech. Rep. SYCON92-01, Ruters Center for Systems and Control, 1992.

[12] Zhang, J. and J. Modestino "The Mean-field Theory in EM Procedures for Markov Random Fields," *International Symposium of Information Theory*, 1991.

The Complexity of Learning from a Mixture of Labelled and Unlabelled Examples

Joel Ratsaby and Santosh S. Venkatesh
Department of Electrical Engineering
University of Pennsylvania
Philadelphia, PA 19104

Abstract

We investigate the tradeoff between labelled and unlabelled sample complexities in learning a classification rule for a parametric two-class problem. In the problem considered, a sample of m labelled examples and n unlabelled examples generated from a two-class, N-variate Gaussian mixture is provided. If side information specifying the parametric form of the probability densities is provided to the learner, maximum likelihood estimation of the unknown class means and *a priori* class probabilities leads to a misclassification probability which deviates from the minimal Bayes error rate by $\mathcal{O}(N^{3/5}n^{-1/5}) + \mathcal{O}(e^{-c_1 m})$ where c_1 is a positive constant. Under weaker side information assumptions, kernel density estimation of the mixture density leads to a misclassification probability which deviates from the minimal Bayes error rate by $\mathcal{O}(N^{c_2 \log N} n^{-2\log N/N}) + \mathcal{O}(e^{-c_1 m})$ where c_1 and c_2 are positive constants.

1 Introduction

The classical problem of *learning* a classification rule can be stated as follows: patterns from classes "1" and "2" (or "states of nature") appear with probabilities $p_1 = p$ and $p_2 = 1 - p$, respectively; the pattern classes are represented by feature vectors x in a common N-dimensional Euclidean space $\mathbb{R}^N$, the patterns of class "i" distributed according to the class-conditional probability density $f_i(x)$ $(i = 1, 2)$. Labelled pairs $(x, y) \in \mathbb{R}^N \times \{1, 2\}$ are assumed generated according to the following mechanism: a pattern class (or "label") $y \in \{1, 2\}$ is first drawn randomly according to the distribution of classes $\{p_1, p_2\}$; a corresponding random feature vector $x \in \mathbb{R}^N$ is then drawn according to the class-conditional density f_y. In the supervised learning scenario, a *labelled* m-sample $\{(x_j, y_j), 1 \le j \le m\}$ is acquired by independent sampling from the distribution of pairs (x, y). Using the sample, the objective is to construct a decision rule which when presented with a random pattern x drawn from the mixture density

$$f(x) = p_1 f_1(x) + p_2 f_2(x)$$

produces a label which disagrees with the true class of origin by a probability P_{error} close to the minimal P_{Bayes} error rate. Formally this learning problem can be formulated in the framework of the Probably Approximately Correct (PAC) learning model (cf. [9, 10])

as follows: Given $\epsilon > 0$, $\delta > 0$, and a labelled sample of size $m = m(\epsilon, \delta)$, construct a classification rule which with confidence in excess of $1 - \delta$ has an error probability

$$P_{\text{error}} = P_{\text{Bayes}}(1 + c_0 \epsilon).$$

(Here and in the sequel, c_0, c_1, c_2, ... represent positive constants independent of the error and confidence parameters ϵ and δ and the dimension of the feature space N.)

As is well known in the unsupervised learning literature (cf. [1]), an *unlabelled* teaching sample $\{x_i \in \mathbb{R}^N : 1 \leq i \leq n\}$ drawn by independent sampling from the mixture density $f(x)$, can also be used in the process of learning classification. If the mixture probability density function

$$f(x \mid \theta) = p_1 f_1(x \mid \theta_1) + p_2 f_2(x \mid \theta_2)$$

is parametric (with unknown parameters $\theta = [\theta_1, \theta_2]$ and p) and is identifiable (cf. Teicher [8]) then the unlabelled sample $\{x_i\}$ can be used to generate an identifiable density function $f\left(x \mid \hat{\theta}\right)$ as an estimate of $f(x \mid \theta)$. Identifiability ensures that the Bayes optimal decision border $\{x : p_1 f_1(x \mid \theta_1) = p_2 f_2(x \mid \theta_2)\}$ can be deduced if $f(x \mid \theta)$ is known and therefore one can construct an estimate of the Bayes border by using $f\left(x \mid \hat{\theta}\right)$ instead of $f(x \mid \theta)$ (such plug-in rules are asymptotically optimal cf. Glick [11]). Once the decision border is estimated, a *small* labelled sample suffices to learn (with high confidence and small error) the appropriate class labels l_1, $l_2 \in \{1, 2\}$ associated with the two disjoint decision regions in $\mathbb{R}^N$ generated by the estimate of the Bayes decision border.

Clearly, unlabelled examples, albeit of less worth than labelled examples, can still carry information of value to the learner. Furthermore, in many cases of practical interest, unlabelled examples exist in profusion in nature (or are inexpensively acquired) while their labelled counterparts are few in number (or are expensive to acquire). For example, in a two-class tree recognition problem "Douglas Fir" and "Spruce," unlabelled examples of these trees may exist in profusion in a forest but labelling them requires the services of a human expert who charges by the hour. Or, in an (idealised) cancer diagnosis scenario, obtaining X-rays of cells may be relatively cheap compared to the cost of the expert labelling them. A mixed sample learning system utilising both labelled and unlabelled examples may hence be of interest where one trades off expensive labelled examples for (very many) unlabelled examples. In such a scenario, the learner would like to determine the exact tradeoff between labelled and unlabelled examples. This question was first posed by T. M. Cover [4] in the following succinct but evocative form: *How many unlabelled examples is each labelled example worth*? In recent work Castelli and Cover [2] have shown that if an infinity of unlabelled examples is available then for identifiable classes with p, f_1 and f_2 unknown, the probability of error decreases exponentially fast in the number of labelled examples to the Bayes risk; in related work [3] they have also shown that if only a finite number of labelled and unlabelled examples are available but the class-conditional densities f_1 and f_2 are known with the only unknown being the mixing parameter p, then under suitable regularity conditions labelled samples are exponentially more valuable than unlabelled samples.

We begin in Section 2 with the purely labelled case where all the examples are labelled: $m < \infty$, $n = 0$. The results for this case provide a useful point of comparison for the mixed sample case.

In Section 3, we consider a parametric situation where the class-conditional densities $f_i(x \mid \theta_i)$ ($i = 1, 2$) are specified by a parameter (vector) $\theta_i \in \mathbb{R}^N$. We assume that the learner is provided with a finite mixed sample of m labelled examples and n unlabelled examples together with side-information specifying the parametric *form* of the class-conditional densities (but not the densities themselves); the parameter vector $\theta = [\theta_1, \theta_2]$ as well as the mixing parameter p are, however, unknown to the learner. Our goal is to determine how the error rate depends on the sample sizes m and n and on the dimensionality N. The results are not immediate even for the archetypal problem of two N-dimensional Gaussian distributed pattern classes and for definiteness we will focus on the case of Gaussian mixtures in this paper. The analysis techniques, however, pertain to other parametric families as well though the rates and constants depend upon the family under consideration.

For conciseness, we limit our discussion in the main body of the paper to a mixture of multi-dimensional Gaussians with unit covariance matrices

$$f_i(x \mid \theta_{0i}) = (2\pi)^{-N/2} e^{-\frac{1}{2}(x-\theta_{0i})^T(x-\theta_{0i})}, \qquad i = 1, 2.$$

To keep things simple we sketch the results for the case where the mixing parameter p is identically $1/2$ and is known to the learner. The mixture density is then just

$$f(x \mid \theta_0) = \tfrac{1}{2} f_1(x \mid \theta_{01}) + \tfrac{1}{2} f_2(x \mid \theta_{02}).$$

We will suppose that the learner knows the form of f but does not know the true parameter vector $\theta_0 = [\theta_{01}, \theta_{02}] \in \mathbb{R}^{2N}$. The results for the general case where the mixing parameter $p \in [0, 1]$ is also unknown to the learner are summarised at the end of the paper.

In Section 4, we consider weaker side information where it is only known that the mixture density belongs to a rich family of classification problems which have the property that the Bayes decision border is uniquely identified by the modes of the corresponding mixture density.

We conclude in Section 5 with a discussion of the specific trade-offs that are achievable in labelled and unlabelled sample complexity, and the effects of dimensionality.

2 Moment Estimation

Given a sample $\{(\xi_j, y_j) \in \mathbb{R}^N \times \{1, 2\} : 1 \leq j \leq m\}$ of m labelled examples obtained by independent sampling from the marginal distribution on pairs (x, y), it is easy to see that in the lack of any further information, moment estimation yields optimal results.

Algorithm E For $i = 1$, 2, let

$$\{\xi^i_{j_k} : y^i_{j_k} = i, 1 \leq k \leq m_i\}$$

be the subsequence of examples generated by class "i." (Clearly, $m_1 + m_2 = m$.)

E1. Estimate the unknown mean vector θ_{0i} of class "i" by

$$\hat{\theta}_i = \frac{1}{m_i} \sum_{k=1}^{m_i} \xi^i_{j_k} \qquad (i = 1, 2).$$

E2. Select as separating surface the hyperplane which is orthogonal to the vector $\hat{\theta}_2 - \hat{\theta}_1$ and passes through the point $(\hat{\theta}_1 + \hat{\theta}_2)/2$. (This separating hyperplane is just the empirical estimate of the Bayes decision border.)

E3. Label points in the resulting two regions R_1 and R_2 partitioning $\mathbb{R}^N$ by the subscript "i" of the mean estimate $\hat{\theta}_i$ which they contain.

Simple Chernoff bounds now suffice to prove the following assertion. The proof is omitted.

Theorem 1 *Suppose two equiprobable pattern classes are distributed according to N-variate Gaussian probability densities $f_1(x \mid \theta_{01})$, $f_2(x \mid \theta_{02})$, both with unit covariance matrices and unknown means θ_{01}, $\theta_{02} \in \mathbb{R}^N$. Then, for sufficiently small $\epsilon > 0$ and arbitrary $\delta > 0$, given*

$$m = \frac{4N}{\epsilon} \log\left(\frac{8N}{\delta}\right)$$

labelled examples and $n = 0$ unlabelled examples, with confidence at least $1 - \delta$, Algorithm E results in a decision rule with classification error bounded by

$$P_{\text{error}} \leq P_{\text{Bayes}}(1 + c_0\epsilon),$$

where $c_0 > 0$ is a constant depending only on the distance between the means.

For a given labelled sample size, the minimal error rate varies (roughly) as the inverse of the sample size; the deleterious effects of dimensionality are evidenced in the $N \log N$ term.

We now turn to the mixed sample case and our main results.

3 Maximum Likelihood Estimation

Let us now consider the effect of introducing n unlabelled examples; the reduction in the size m of the labelled sample needed to obtain a given error rate will yield a rough measure of the worth of each labelled example in terms of the number of unlabelled examples that are needed to compensate for it.

Suppose that it is known that the sample is engendered by a Gaussian mixture in which the class conditional densities $f_i(x \mid \theta_{0i})$ $(i = 1, 2)$ are N-dimensional Gaussians with unit covariance matrices and unknown mean vectors $\theta_{0i} \in \mathbb{R}^N$. We assume in addition that the unknown parameter vector $\theta_0 = [\theta_{01}, \theta_{02}]$ lies in a compact set $\Theta \subset \mathbb{R}^{2N}$. Our approach will be to utilise the unlabelled sample to estimate the unknown parameter vector θ_0, hence also the mixture density $f(x \mid \theta_0) = \frac{1}{2}f_1(x \mid \theta_{01}) + \frac{1}{2}f_1(x \mid \theta_{01})$, from which we can then obtain an estimate of the hyperplane which constitutes the Bayes decision border in this case. The labelled sample will then be utilised to provide a labelling of the two disjoint regions in $\mathbb{R}^N$ induced by the separating hyperplane.

Algorithm M Let $\{x_i, 1 \leq i \leq n\}$ be the unlabelled n-sample obtained by independent sampling from the mixture density $f(x \mid \theta_0)$, and let $\{(\xi_j, y_j), 1 \leq j \leq m\}$ denote the labelled m-sample obtained by independent sampling from the distribution of pairs (x, y).

M1. Estimate the unknown parameter vector θ_0 by the maximum likelihood estimate

$$\hat{\theta} = \arg\sup_{\theta \in \Theta} \tfrac{1}{n} \sum_{i=1}^{n} \log f(x_i \mid \theta).$$

M2. Select as separating surface the hyperplane that passes through the point $(\hat{\theta}_1+\hat{\theta}_2)/2$ and is orthogonal to the vector $\hat{\theta}_2 - \hat{\theta}_1$.

M3. Label each of the two decision regions separated by the hyperplane according to the majority of the labelled examples in the region.

The following is indicative of the kind of tradeoff between labelled and unlabelled examples that is possible in this setting.

Theorem 2 *Suppose two equiprobable pattern classes are distributed according to* N*-variate Gaussian probability densities* $f_1(x \mid \theta_{01})$, $f_2(x \mid \theta_{02})$, *both with unit covariance matrices and unknown means* $\theta_{01}, \theta_{02} \in \mathbb{R}^N$, *and suppose that the unknown parameter vector* $\theta_0 = [\theta_{01}, \theta_{02}]$ *lies in a compact subset* Θ *of* $\mathbb{R}^{2N}$. *Then for sufficiently small* $\epsilon > 0$ *and arbitrary* $\delta > 0$, *given*

$$m = c_1 \log \delta^{-1}$$

labelled examples and

$$n = \frac{c_2 N^2}{\epsilon^3 \delta}\left(N \log \epsilon^{-1} + \log \delta^{-1}\right)$$

unlabelled examples, with confidence in excess of $1-\delta$, *Algorithm M determines a decision rule with classification error bounded by*

$$P_{\text{error}} \leq P_{\text{Bayes}}(1 + c_3\epsilon).$$

In the above, c_1, c_2, *and* c_3 *are positive constants which are independent of* ϵ, δ, N, *and the choice of* $\theta_0 \in \Theta$.

The main technical tool needed in the proof is a rate of convergence in a uniform strong law for bounded functions (cf. Pollard [6, Theorem II.6.37] and Haussler [5]). The messy technical considerations involve truncation arguments to exploit the exponential decay of the Gaussian to massage the problem into a form where the uniform strong law is applicable.

4 Kernel Density Estimation

Under weaker side information, the sample complexity demands increase dramatically. Consider, for instance, a rich family of classification problems (which include Gaussian mixtures) for which the Bayes border is identified solely by the modes of the mixture. The approach here uses kernel estimation combined with a mode-estimation algorithm with which the Bayes decision rule is estimated. We briefly sketch the algorithm. (For the technical details pertaining to the problem class and the algorithm, see [7].)

Algorithm K Let $\{x_i, 1 \leq i \leq n\}$ be the unlabelled n-sample obtained by independent sampling from the mixture density $f(x \mid \theta_0)$, and let $\{(\xi_j, y_j), 1 \leq j \leq m\}$ denote the labelled m-sample obtained by independent sampling from the distribution of pairs (x, y).

K1. Use kernel estimation to obtain the mixture density estimate f_n from the unlabelled sample. (It is not necessary for f_n to be a *bona fide* density.)

K2. Determine a set of modes of f_n which are consistent estimators of the modes of the underlying mixture f.

K3. Let the decision border be a hyperplane which passes through the average of the mode estimates and is perpendicular to their least-square line. (In the Gaussian case there are two mode estimates.)

K4. Label the two decision regions induced by the hyperplane by the label of the majority of the examples in each region.

The following sample complexity estimates for Algorithm K also arise by careful application of uniform strong laws.

Theorem 3 *Suppose we are given two classes which are distributed according to Gaussian probability densities* f_1*,* f_2*, with means* θ_{01}*, and* θ_{02} *respectively, and with unit covariance matrices. Suppose further that* $0 < P_{Bayes} \leq 0.16$ *(or, equivalently* $|\theta_{01} - \theta_{02}| > 2$*). Then there exists a positive constant* b *(determined by* $|\theta_{01} - \theta_{02}|$*) such that for* $0 < \epsilon < b$ *and arbitrary* $\delta > 0$*, given*

$$n = \frac{c_4 13^{N \log N + N \log 5}}{\epsilon^{\frac{N}{2 \log N}}} \log\left(\epsilon^{-1}\delta^{-1}\right)$$

unlabelled examples and

$$m = c_5 \log \delta^{-1}$$

labelled examples, Algorithm K determines a decision rule with a classification error

$$P_{\text{error}}(m, n) \leq P_{\text{Bayes}}(1 + c_6\epsilon)$$

with confidence at least $1 - \delta$*, where* $c_4 > 0$ *an absolute constant,* $c_5 > 0$ *is a constant depending on* P_{Bayes}*, and* $c_6 > 0$ *depends on* θ_0*.*

REMARKS: The restriction on P_{Bayes} is a consequence of the constraint on the two means of the class conditional densities to be sufficiently distant in order for the mixture to have two modes and thereby identify the Bayes border.

Based on Algorithm K, a more general statement can in fact be made pertaining to every problem in a regular family of mixtures whose modes identify the Bayes border. Indeed, the sample complexities specified in Theorem 3 will hold uniformly for all problems in this class.

5 Conclusions

For the mixture Gaussian problem, a classification error probability of $P_{\text{Bayes}}\left(1 + O(\epsilon)\right)$ can be obtained with confidence in excess of $1 - \delta$ given a random mixed sample of

$$n_M = \frac{c_8 N^2}{\epsilon^3 \delta}\left(N \log \epsilon^{-1} + \log \delta^{-1}\right)$$

unlabelled examples and

$$m_M = c_9 \log \delta^{-1}$$

labelled examples. Compare this with the purely-labelled sample scenario of Section 2 where a learner using Algorithm E requires

$$m_E = \frac{4N}{\epsilon} \log\left(\frac{8N}{\delta}\right)$$

labelled examples to achieve the same learning performance. Clearly, the introduction of unlabelled examples has reduced the demands on the number of labelled examples. As a rough measure one may take the ratio

$$\frac{n_M}{(m_E - m_M)} = \frac{c_{10} N^2}{\epsilon^3 \delta \log N}$$

as indicative of the number of unlabelled examples needed to compensate for the removal of each labelled example. In this sense, the tradeoff is polynomial in N, $\frac{1}{\epsilon}$, and $\frac{1}{\delta}$. Note that, as claimed in the abstract, inverting the sample complexity estimates for the mixed sample case shows that the error rate contribution from the labelled examples decreases exponentially fast in m, while the error rate contribution from the unlabelled examples decreases only as an inverse polynomial in n. Note also that the efficacy of the unlabelled examples in the algorithmic scenarios discussed here diminishes as the dimensionality N increases. More explicitly, the classification error probability under Algorithm M deviates from the Bayes optimal error rate by $\mathcal{O}\left(N^{3/5} n^{-1/5}\right) + \mathcal{O}\left(e^{-cm}\right)$.

Note that the analysis will work for other parametric families as long as the tails of the distributions are not too heavy. In particular, the analysis works *in toto* for other members of the exponential family,, albeit with different constants.

The case of a general unknown mixing parameter p is also easily handled in this framework. Two modifications to the approaches are indicated. In the first approach, the labelled examples can be used to directly estimate p according to the relative frequency of class "1;" Chernoff bounds show that very good estimates of p can be obtained from a small labelled sample. In the second approach, p is treated as an additional unknown parameter and maximum likelihood estimation is used to estimate p as well as the parameter vector θ_0. If p is bounded away from 0 and 1, a similar analysis shows that each labelled example has to be replaced by of the order of

$$\frac{c_{11} N^2 \log \epsilon^{-1}}{\epsilon^4 p^{11} \log N}$$

unlabelled examples to achieve the same performance.

In nonparametric cases, the tradeoff between labelled and unlabelled examples is much more pronounced as seen in Section 4. For instance, if Kernel density estimation is used to infer the mixture density of an identifiable family of bimodal mixture densities (including Gaussian mixtures), each labelled example is worth roughly

$$\frac{c_{12} 13^{N \log(5+\log N)}}{N \log N \epsilon^{\frac{N}{2 \log N}}}$$

unlabelled examples.

Acknowledgements

This research was supported by the Air Force Office of Scientific Research under grant F49620–93–1–0120. Joel Ratsaby was partially supported by a US Air Force graduate fellowship during the completion of this work.

References

[1] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York, 1973.

[2] V. Castelli and T. M. Cover, "Classification rules in the unknown mixture parameter case: relative value of labelled and unlabelled examples," *Proc. 1994 IEEE Int. Symp. Inform. Theory*, p. 111, Trondheim, Norway, 1994.

[3] V. Castelli and T. M. Cover, "On the exponential value of labelled samples," to appear *Pattern Recognition Letters.*

[4] T. M. Cover, "Learning and generalisation," in *Proc. 4th Annual Workshop on Computational Learning Theory*, (eds. L. G. Valiant and M. K. Warmuth), p. 3, Morgan Kaufmann, San Mateo, California, 1991.

[5] D. Haussler, "Decision theoretic generalisations of the PAC model for neural net and other learning applications," *Technical Report: University of California, Santa Cruz*, UCSC–CRL–91–02, 1989.

[6] D. Pollard, *Convergence of Stochastic Processes*, Springer Verlag, New York, 1984.

[7] J. Ratsaby, *The Complexity of Learning from a Mixture of Labelled and Unlabelled Examples*, Ph.D. Thesis, University of Pennsylvania, 1994.

[8] H. Teicher, "Identifiability of finite mixtures," *Annals of Mathematical Statistics*, vol. 34, pp. 1265–1269, 1963.

[9] L. G. Valiant, "A Theory of the learnable," *Comm. ACM*, vol. 27, no. 11, pp. 1134–1142, 1984.

[10] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth, "Learnability and the Vapnik-Chervonenkis dimension," *JACM*, vol. 36, no. 4, pp. 929–965, 1989.

[11] N. Glick, "Sample-based classification procedures derived from density estimators," *J. American Statistical Association*, vol. 67, 1972.

[12] S. J. Yakowitz and J. D. Spragins, "On identifiability of finite mixtures," *Annals of Mathematical Statistics*, vol. 39, pp. 209–214, 1968.

Addresses

Joel Ratsaby is currently with the Department of Electrical Engineering, Technion, Israel. He may be contacted by e-mail at `jer@ee.technion.ac.il`. Santosh S. Venkatesh may be contacted by e-mail at `venkatesh@ee.upenn.edu`.

Temporal Difference Learning Applied to Multihypothesis Sequential Detection

Chengan Guo & Anthony Kuh *
Department of Electrical Engineering
University of Hawaii at Manoa
Honolulu, Hawaii 96822
chengan@spectra.eng.hawaii.edu & kuh@wiliki.eng.hawaii.edu

Abstract

This paper presents a nonparametric neural network method for multihypothesis sequential detection. We first examine a multihypothesis sequential detection (multi-SPRT) method and come up with performance functions that are used in a neural network. The neural network is trained using the Temporal Difference (TD) learning algorithm, to perform multihypothesis sequential detection with performance that is close to the multi-SPRT. Simulations are conducted showing that for the same performance levels, the multi-SPRT and the neural network method require less than half the observation samples of fixed sample size detectors.

---

*This work was supported in part by NSF grant EET-8857711.

1 Introduction

This paper considers a nonparametric method for multihypothesis sequential detection. We extend the work of [3] by training a neural network using reinforcement learning to perform sequential detection for multiple hypotheses. The neural network consists of a feedforward network with context units that is trained using an extended version of the *Temporal Difference* (TD) learning algorithm developed by Sutton [5]. Unlike the two hypotheses case, where the optimal Sequential Probability Ratio Test (SPRT) designed by Wald [7] is easily implemented, optimal tests cannot be implemented for more than two hypotheses [1, 2]. We compare our neural network sequential detector to a good suboptimal sequential detector, the (multi-SPRT) that is a generalization of the SPRT. Simulation tests on Gaussian data show that the neural network scheme has performance that is comparable to the multi-SPRT and both methods are far better than Fixed Sample Size (FSS) detectors. The neural network method has an additional advantage in that it is a nonparametric method that works well when minimal information is given to the neural network.

Sequential detection involving three or more hypotheses is considerably more difficult than the two hypothesis case [1, 2]. Ghosh and Sen [2] pointed that in general, optimal tests can not be found when there are more than two hypotheses. Research in multihypothesis detection focuses on extensions or generalizations of the SPRT to the case of more than two hypotheses. There has been little research on considering optimal tests, but Baum and Veeravalli [1] design the MSPRT which is a good approximation to the Bayesian solution. Although the generalized SPRT methods are not optimum, they still retain advantages over FSS tests in achieving comparable performance with considerable savings in the number of observation samples. In addition, most multi-SPRT detection methods has simple structure and can be easily realized for parametric models. These schemes have been used in a variety of applications [1, 2].

When distribution statistics are not available the parametric methods must be modified. We will consider using a neural network to perform sequential detection for the multihypothesis case. The neural network is trained with sets of sequences of observation samples to learn the relevant information needed to make good decisions. We have demonstrated in [3] that the neural network can effectively perform sequential detection for the two hypothesis case. This paper generalizes results to the multihypothesis case.

The neural network method is based on the multi-SPRT method which is developed in Section 2 and is equivalent to the MSPRT discussed in [1]. Similar to [3], we develop a set of performance functions that the neural network attempts to learn. The multi-SPRT performance functions are the a posteriori conditional probabilities that an observation sequence is drawn from a hypothesis H_j, given that the decision is H_j, $j = 1, \ldots, M$.

Section 3 focuses on the learning algorithms for the neural network sequential detector. We find that for the sequential detection problem, supervised learning algorithms such as the error back propagation (BP) algorithm [4] are not suitable as discussed in [3]. However, a reinforcement learning algorithm, Sutton's TD learning algorithm [5] is well suited to perform our learning task. We use context units and a modified TD learning algorithm using the error backpropagation algorithm for our multilayer feedforward network. The reinforcement signal consists of an M-vector signal that gives the probability that each hypothesis occurs. Finally, Section 4 presents simulation results and Section 5 summarizes the paper with directions for further research.

2 A Multihypothesis Sequential Probability Ratio Test

2.1 Multi-SPRT Method

Most previous research on sequential multihypothesis detection extends and/or generalizes the SPRT method to the case of more than two hypotheses. A recent survey on this topic is given in [2]. We consider an extended SPRT method for testing multi-hypotheses that is used in our neural network model.

Let $\{X_k, k = 1, 2, \cdots\}$ be a sequence of iid random variables and let H_j represent hypothesis j, $1 \leq j \leq M$ where $M \geq 2$. Define the log-likelihood ratio $Z_j(\mathcal{X}_n)$ for $j = 1, 2, \cdots, M$ by

$$Z_j(\mathcal{X}_n) = \log \frac{p_j f_n(\mathcal{X}_n \mid H_j)}{\sum_{i \neq j}^{M} p_i f_n(\mathcal{X}_n \mid H_i)} = \log \frac{p_j \prod_{k=1}^{n} f(X_k \mid H_j)}{\sum_{i \neq j}^{M} p_i \prod_{k=1}^{n} f(X_k \mid H_i)}. \qquad (1)$$

where $\mathbf{X_n} = (X_1 \ldots, X_n)$, $\mathcal{X}_n = (x_1, \ldots, x_n)$, $f_n(\mathcal{X}_n \mid H_i)$ is the conditional joint density function of $\mathbf{X}_n$ given H_i, and p_i is the prior probability $p_i = P(H_i)$ $(i = 1, \cdots, M)$.

We define detection boundaries, b_j, and the approximation boundaries b'_j by

$$b_j \leq \log \frac{p_j P(D = H_j \mid H_j)}{\sum_{i \neq j}^{M} p_i P(D = H_j \mid H_i)} = b'_j, \quad j = 1, 2, \cdots, M. \qquad (2)$$

where D is a decision rule.

Then the multi-SPRT procedure is defined by (initially set n=1)

$$\begin{array}{ll} (i). & accept\ H_j\ and\ stop,\ if\ Z_j(\mathcal{X}_n)\ \geq b_j\ for\ any\ j\ (j=1,2,\cdots,M); \\ & (\ accept\ H_{j^*}\ with\ j^* = argmax_j\{Z_j(\mathcal{X}_n)/b_j\}\ if\ more\ than\ one\ j \\ & \quad satisfies\ Z_j(\mathcal{X}_n) \geq b_j,\ \ at\ time\ n\ ); \\ (ii). & otherwise\ continue\ sampling\ by\ observing\ X_{n+1}, n \leftarrow n+1\ \ and\ \ goto\ (i) \end{array} \tag{3}$$

The sequential test method is applicable to any number of hypotheses and reduces to the SPRT when there are two of hypotheses. This method is equivalent to the MSPRT proposed by Baum and Veeravalli [1]. Note that $Z_j(\mathcal{X}_n)\ \geq b_j$ of (3) is equivalent to the inequality

$$\frac{p_j f_n(\mathcal{X}_n \mid H_j)}{\sum_{i=1}^{M} p_i f_n(\mathcal{X}_n \mid H_i)} \geq \frac{1}{1+e^{-b_j}}\,. \tag{4}$$

Using (4) and setting $e^{-b_j} = A_j$ results in the MSPRT decision rule. Therefore this multi-SPRT shares all the properties of the MSPRT. We now develop a set of sigmoidal type performance functions from the multi-SPRT that will be learned by the neural network.

2.2 Performance Function for the Multi-SPRT

The performance measurement functions are denoted by $F_j(b_1,\cdots,b_M,n)$ and defined as the posterior probability of correct acceptance of H_j given that $D_n = H_j$, i.e.,

$$F_j(b_1,\cdots,b_M,n) = P(H_j\ true \mid D_n = H_j), \qquad j = 1,\cdots,M. \tag{5}$$

where D_n is the decision rule based on the first n observations. These set of performance functions are closely related to the input/output characteristics of the multi-SPRT and the asymptotic performance functions ($n \to \infty$) are the functions that are learned by the neural network sequential detector.

Following [3], we establish the asymptotic behavior and find lower bounds for the performance functions.

Theorem 1 *Assume that the prior probabilities, $p_j > 0$ for $j = 1,\cdots,M$. Then the asymptotic performance function $F_j(b_j) = \lim_{n\to\infty} F_j(b_1,\cdots,b_M,n)$ is a sigmoidal function of b'_j and is lower bounded with*

$$F_j(b_j) = \frac{1}{1+e^{-b'_j}} \geq \frac{1}{1+e^{-b_j}}, \qquad j = 1,2,\cdots,M. \tag{6}$$

Proof: By Bayes rule the performance function can be represented by:

$$F_j(b_1, \cdots, b_M, n) = \frac{p_j P(D_n = H_j \mid H_j)}{\sum_{i=1}^{M} p_i P(D_n = H_j \mid H_i)} = \frac{1}{1 + \frac{\sum_{i \neq j}^{M} p_i P(D_n = H_j \mid H_i)}{p_j P(D_n = H_j \mid H_j)}}, \quad j = 1, \cdots, M. \tag{7}$$

Following arguments in [3] we have that

$$\lim_{n \to \infty} P(D_n = H_j \mid H_i) = P(D = H_j \mid H_i), \quad \forall\, i, j = 1, \cdots, M. \tag{8}$$

Then from (7) and (2), we get that

$$\begin{aligned} \lim_{n \to \infty} F_j(b_1 \cdots, b_M, n) &= \lim_{n \to \infty} \frac{1}{1 + \frac{\sum_{i \neq j}^{M} p_i P(D_n = H_j \mid H_i)}{p_j P(D_n = H_j \mid H_j)}} \\ &= \frac{1}{1 + \frac{\sum_{i \neq j}^{M} p_i P(D = H_j \mid H_i)}{p_j P(D = H_j \mid H_j)}} \\ &= \frac{1}{1 + e^{-b'_j}} \geq \frac{1}{1 + e^{-b_j}}, \quad j = 1, \cdots, M. \end{aligned} \tag{9}$$

Note that the multi-SPRT procedure is implemented by comparing the log-likelihood ratios $Z_j(\mathcal{X}_n)$ with the detection boundary b_j. When $Z_j(\mathcal{X}_n) = b_j$, then $F_j(b_j)$ evaluated for that particular log-likelihood ratio will give the a posteriori correct detection probability $P(H_j \mid D : \mathcal{X}_n \in H_j)$. Therefore, $F_j(b_j)$ is able to describe the relationship between $P(H_j \mid D = H_j)$ and $Z_j(\mathcal{X}_n)$s. This input / output characteristic is used in the design of a neural network sequential detector.

3 Neural Network Sequential Detection

In this section we summarize the main ideas for using the TD learning algorithm to perform multihypothesis sequential detection when observations are iid. This work extends [3] from the two hypotheses to the multi-hypotheses case.

3.1 Neural Network Multihypothesis Sequential Detection Algorithm

We first assume that the log-likelihood ratio $Z_j(\mathcal{X}_n)$s can be learned by a neural network. The output units of the neural network are then sigmoidal units with the outputs set to $y_j(n) = 1/(1 + e^{-Z_j(\mathcal{X}_n)})$. The neural network sequential detection algorithm is then described by (initially set t=1):

$$\begin{array}{rl} (i). & \textit{input } x_t \textit{ and compute } Z_j(\mathcal{X}_t)\ (j=1,\cdots,M); \\ (ii). & \textit{check outputs to see if any } y_j(t)\ge p_{jj}\ (j=1,\cdots,M): \\ & \bullet\ \textit{decide } D=H_j \textit{ if only one output satisfies this relation },\\ & \qquad\qquad D = H_{j^*} \textit{ with } j^*=argmax_j\{y_j(t)/p_{jj}\} \textit{ if more than one} \\ & \qquad\qquad\qquad y_j(t) \textit{ satisfies this relation , and stop };\\ & \bullet\ \textit{otherwise, } t\leftarrow t+1 \textit{ and goto } (i)\ . \end{array} \tag{10}$$

where p_{jj} $(j=1,\cdots,M)$ are predetermined performance bounds.

Noting that $\frac{p_j f_n(\mathcal{X}_n|H_j)}{\sum_{i=1}^{M} p_i f_n(\mathcal{X}_n|H_i)} = \frac{1}{1+e^{-Z_j(\mathcal{X}_n)}}$ and using (4), we see that the algorithm described by (10) is the same as the multi-SPRT algorithm (3) when boundaries are matched with

$$p_{jj} = \frac{1}{1+e^{-b_j}}, \qquad j=1,\cdots,M. \tag{11}$$

3.2 Applying TD Learning to the Neural Network

We now discuss how the neural network learns the the log-likelihood ratio function $Z_j(\mathcal{X}_n)$s. Our neural network is a nonparametric model that does not require log-likelihood ratios. We assume that during training, an M-vector is available at the termination time of each observation sequence to indicate the correct hypothesis from which the sequence was drawn. As stated in [3], supervised learning such as the error back propagation (BP) algorithm cannot be effectively used for sequential detection. We find, however, that a reinforcement learning algorithm, the TD learning algorithm, [5] is well suited to sequential detection. The TD learning assigns credits using the differences between the temporal successive predictions of the target instead of using the differences between the prediction and the target signal. The reinforcement signal is the M-vector available at the termination of each sequence.

Figure 1 in Section 3.3 shows a block diagram of the neural network and the reinforcement learning procedure applied to the sequential detection problem. The input x_t is an observation drawn at time t from an observation sequence $\mathcal{X}_n$. A reinforcement signal vector $R(t) = [r_1(t),\cdots,r_M(t)]^T$ is given for each input. The outputs $Y(t) = [y_1(t),\cdots,y_M(t)]^T$ are used to learn the performance functions $[F_1(Z(\mathcal{X}_t)),\cdots,F_M(Z(\mathcal{X}_t))]^T$.

The TD learning algorithm applied to sequential detection is then described as follows. As each input observation x_t is fed into the network sequentially, the network generates an output $Y(t)$ which is a prediction for the final reward signal $R(n)$. Then the successive prediction errors $E(t) = Y(t+1) - Y(t)$ $(t = 1,2,\cdots,n-1)$ and $E(n) = R(n) - Y(n)$ are used to update the weights of the network by using the TD learning algorithm update rule

$$\Delta W_m(t) = E(t) \sum_{k=1}^{t} \lambda^{t-k} \nabla_w Y(k) , \qquad W_{m+1} = W_m + \mu \sum_{t=1}^{n} \Delta W_m(t) \tag{12}$$

where λ is the parameter of TD learning algorithm, μ is learning rate, and $\nabla_w Y_k$ is the gradient of $Y(k)$ with respect to W_m . This algorithm is used to update the weights of the output layer. The updating rule then uses the error backpropagation algorithm to update weights of hidden layer units.

In the learning process, $R(t)$ plays the evaluative role in the reinforcement learning to guide the update of the network weights for each observation sequence. Note that, for the sequential detection problem, $R(t)$ is the M-vector that is available only at time n, the termination time of the sequence. We set $R(n) = [p_{1j}, \cdots, p_{jj}, \cdots, p_{Mj}]^T$ when the training sequence $\mathcal{X}_n$ is from H_j $(j = 1, \cdots, M)$, where

$$\sum_{i=1}^{M} p_{ij} = 1 , \qquad p_{ij} \geq 0 , \qquad i, j = 1, \cdots, M. \tag{13}$$

At other times, $(t \neq n)$, $R(t)$ is not available.

3.3 Architectures for the Neural Network

There is one more crucial problem we must address before the neural network can perform sequential detection. How can we go from input observations X_t, $1 \leq t \leq n$ to a sufficient statistic where decisions can be made (e.g., the log-likelihood ratio $Z_j(\mathcal{X}_n)$). We must somehow keep track of all the input observations as they are input into the network. We would like to train a feedforward network using the TD learning algorithm and somehow store the dynamic information that is needed for a sequential detection procedure. This is accomplished by using context units which will store the dynamic information and then using a feedforward network where the TD learning algorithm is applied. To see how we can construct this neural network, let us examine the conditional density functions where each observation is chosen from. Many common random variables used in engineering applications such as the Gaussian, Poisson, Exponential, and Bernoulli random variables belong to the exponential family with linear sufficient statistics [2]:

$$f(x; \theta) = h(x) e^{C(\theta) \cdot x + D(\theta)} , \tag{14}$$

where $h(x)$ is a function of only x (the observation),and $C(\theta)$ and $D(\theta)$ are functions of only θ (the hypothesis parameters). In this case the log-likelihood ratio $Z_j(\mathcal{X}_t)$ has the form

$$Z_j(\mathcal{X}_t) = \log \frac{p_j \prod_{k=1}^{t} f(X_k \mid H_j)}{\sum_{i \neq j}^{M} p_i \prod_{k=1}^{t} f(X_k \mid H_i)} \triangleq G_j(\sum_{k=1}^{t} X_k,\ t) . \tag{15}$$

Equation (15) shows that $Z_j(\mathcal{X}_t)$ is a function of $\sum_{k=1}^{t} X_k$ and t. From this result we construct a neural network architecture shown in Figures 1 for the neural network sequential detector that is suitable to learn random sequences that satisfy (14). In Figure 1 the units marked with " T " represent a unit time delay operation and the input nodes marked with "+" are summation units. We call these units context units. We use these context units to store the cumulative information $\sum_{k=1}^{t} X_k$, and the sample number t of the sequence. The feedforward network in Figure 1 is a multilayer neural network that learns the log-likelihood ratios with sigmoidal output units. The overall neural network therefore learns each of the M performance functions. with the weights of the feedforward network learned via the TD learning algorithm given in Section 3.2 .

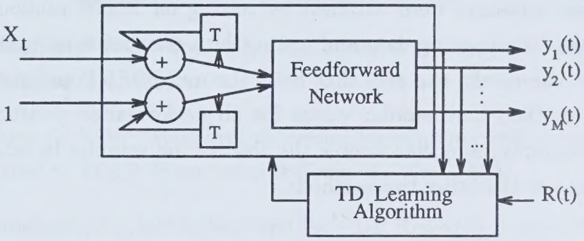


Figure 1. A TD learning network for sequential multihypothesis detection.

This neural network can easily be modified to handle distributions with nonlinear sufficient statistics and even observation data that is not iid.

4 Simulations

Many simulation experiments have been conducted for the multi-SPRT, the TD learning network sequential detection method, and also for the best fixed-sample-size (FSS) detection method [6]. In the simulations, the distribution function and parameters to be detected are given for the multi-SPRT and FSS detection methods. This statistical information is not available to the neural network as only the M-vector reinforcement signal is available. Simulations are presented for Gaussian data (referred to as the standard example by [2]) with three hypotheses. Each hypothesis generates iid Gaussian sequences with variance one and different means (with mean values 1, -1, and 0).

In the simulations, the TD learning parameter is set to $\lambda = 0.5$, the learning rate to $\mu = 0.1$, weights initialized to small random values, and the predetermined correct detection probabilities set to $p_{11} = p_{33} = 0.99$ and $p_{22} = 0.98$. The multi-SPRT is computed under equivalent conditions with its detection boundaries b_j being determined by using (11). All detection methods use the same simulation data.

Table 1 shows the simulation results for the standard example. It gives simulation values of the performance functions $F_j = P(H_j \; true \mid D = H_j)$, the average correct detection rate $\bar{R}$, and the average observation sample number, $\bar{N}$. The values for the network detector were obtained by training the TD learning network on 1000 randomly generated detection sequences. Each run through the 1000 detection sequence constituted one iteration. Table 1 shows results with 200 training iterations. The values for all the performance measures were obtained by testing on 10,000 randomly generated detection sequences. The training data and testing data are chosen independently in the simulations. From the result, one sees that both the multi-SPRT and the TD learning network (TDLN) method have similar values for all performance measures. However, the FSS detector requires more than double the observation samples to achieve the same level of performance as the other two methods.

Table 1: Detection performances of multihypothesis test methods for standard example

	F_1	F_2	F_3	$\bar{R}$	$\bar{N}$
FSS Detection	.9917	.9890	.9926	.9911	25.00
multi-SPRT	.9936	.9890	.9934	.9920	11.72
TDLN	.9936	.9896	.9926	.9919	11.76

We also conducted simulations for a four hypotheses case and simulations involving other distributions. Results were similar to the three hypotheses Gaussian data.

5 Summary

In this paper we studied multi-hypotheses sequential detection. We first introduced a multi-SPRT method for testing multihypotheses. We then used a neural network as a nonparametric sequential detector. By introducing a set of performance measurement functions for the multi-SPRT, we designed a neural network to realize these functions. We found that a reinforcement learning algorithm, the TD learning algorithm was well suited to train the neural network. The neural network is a nonparametric sequential detector consisting of context units and a feedforward network. A number of simulations were conducted showing that the nonparametric neural network method had performance that was almost identical to the multi-SPRT and both methods were far better than FSS detectors.

Further simulations can be conducted on distributions with nonlinear sufficient statistics and observation data that is not iid. It also remains to be shown that the TD learning algorithm converges in this application and we conjecture that the performance is the same as the multi-SPRT method.

References

[1] C. W. Baum and V.V. Veeravalli, "A sequential procedure for multihypothesis testing," *IEEE Trans. Inform. Theory,* vol.40, pp.1994-2007, Nov., 1994.

[2] B. K. Ghosh and P. K. Sen Editors, *Handbook of Sequential Analysis,* New York: Marcel Dekker, Inc., 1991.

[3] C. Guo and A. Kuh. "Temporal difference learning applied to sequential detection ", submitted to *IEEE Trans. Neural Networks,* Nov. 1994.

[4] D. E. Rumelhart, J.L.McClelland and the PDP Research Group, *Parallel Distributed Processing,* Volume 1, MIT Press, Cambridge, Mass., 1986.

[5] R. S. Sutton, "Learning to predict by the methods of temporal differences, " *Machine Learning,* 3, pp. 9-44, 1988.

[6] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part I.* New York: John Wiley & Sons, 1968.

[7] A. Wald, *Sequential Analysis,* New York: John Wiley & Sons, 1947.

[8] A. Wald and J. Wolfowitz, " Optimum character of the sequential probability ratio test," *Ann. Math. Statist. 19,* pp. 326-339, 1948.

A Robust Temporal Neural Network for Blind Channel Identification *

Guojie Dong and Ruey-wen Liu
University of Notre Dame
Notre Dame, IN 46556

Abstract

A temporal neural network is proposed for blind identification of communication channels. This neural network is shown to be globally convergent for almost all initial conditions. It is on-line and needs little memory. Its input is not necessarily stationary and can be intermittent in time. Computer simulation shows that the neural network can track time-varying channels, and can retrieve itself after a sudden change of the channel coefficients. This neural network is especially useful for mobile communication where channels are constantly changing, and possibly nonminimum phase.

I. Introduction

In digital wireless communications, the multipath phenomenon causes intersymbol interference (ISI). In dealing with this problem, equalization technique is used. Conventional equalization technique requires transmitting training signals so that the communication channel can be identified/equalized. However, in wireless environment, the channel is constantly changing. The training signal has to be sent periodically to update the channel estimates, which reduces the transmitting efficiency of the channel. For example, the GSM standard currently adopted in Europe requires 28 training bits per 116 information bits, 20% of total transmitting signals are the training signals. This drawback makes the blind equalization/identification methods very attractive, because they do not require to transmit training sequence.

The blind equalization techniques only use the channel output signals to identify/equalize the channel. Since it is known that the channel can not be identified by using second order statistics of the channel output alone if the channel is one of non-minimal phase, high order statistics were used in many blind identification/equalization approaches [1]-[4]. But this method, as expected, suffers from computation intensity and unreliable estimation of high order statistics. Recently, Tong *et al* [5] have shown that blind identification is also possible by use of only second-order statistics, provided that the received signal is oversampled. Later, it has been shown in [6], [7] and [8] that the same can be achieved by using multiple receivers, and Moulines *et al* [7] showed

---

This work was supported in part by the Office of Naval Research N00014-91-J-1461.

the equivalence between oversampling signal from one receiver and signals from multiple receivers.

In this paper, we propose a temporal neural network to perform the blind channel identification either using oversampled received signals, or using multiple receivers. It is on-line, robust and require little memory. Furthermore, the input signal needs not to be stationary and can be intermittent in time. Simulations show that it is able to track time-varying channels, and to retrieve itself after some sudden change in channels.

II. Problem Statement

Without using high order statistics, blind channel identification can be achieved by either using signals from multiple receivers or using oversampled signals from one receiver. Therefore, for the most simple case, it can be modeled as a two-channel identification model shown in Figure 1, where $s(\cdot)$ is the common input sequence of the channels, whose impulse responses are $\{h_i(\cdot)\}$, $i = 1, 2$, and $x_i(\cdot)$ the output sequence from the i^{th} channel. The temporal neural network is used to identify the channel impulse responses from $x_1(\cdot)$ and $x_2(\cdot)$.

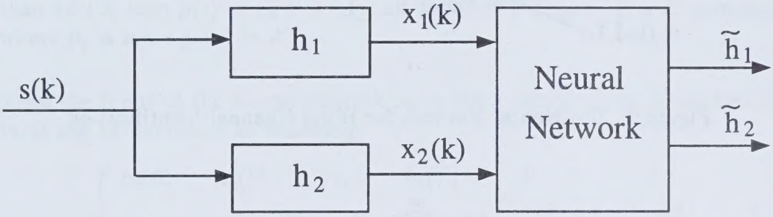


Figure 1: The Model for Blind Channel Identification

The following assumptions are made on the signal and the channels:

1. $s(k)$ is bounded, but not necessarily stationary.
2. Both channels are FIR and their impulse responses are represented by $\{h_i(\cdot)\}$, $i = 1, 2$.

The relations among $x_i(\cdot)$, $s(\cdot)$ and $h_i(\cdot)$ are given by:

$$x_i(k) = \sum_{j=0}^{L} h_i(j)s(k-j) \qquad i = 1, 2 \tag{1}$$

where $L + 1$ is the maximum length of the two channel responses. Denote the vector forms of channel responses by

$$\mathbf{h}_i = (h_i(0), h_i(1), \cdots, h_i(L))^t \qquad i = 1, 2$$

and denote

$$\mathbf{h} = \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \end{bmatrix} \tag{2}$$

Our objective is to design a temporal neural network whose weights converges to **h** up to a scalar. *This shall be accomplished without any statistical information of s.*

III. The Temporal Neural Network for Channel Identification

The proposed temporal neural network to perform blind channel identification is shown in Figure 2:

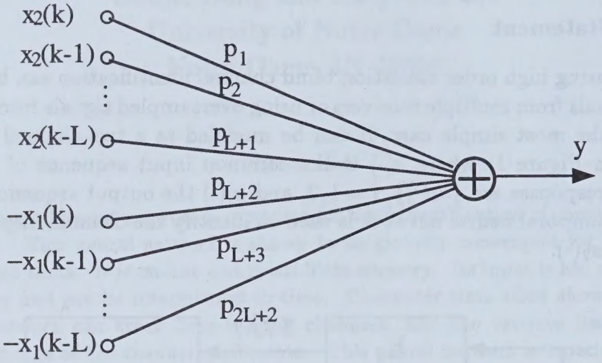


Figure 2: The Neural Network for Blind Channel Identification

where

$$\mathbf{x}(t) = \sum_{k=0}^{\infty} \mathbf{x}_k r(t - kT) \tag{3}$$

is the input of the neural network, $\mathbf{p} \in R^{2L+2}$ is the connection weight, and $y \in R$ is the output of the neural network. In Eq. (3),

$$\mathbf{x}_k = (x_2(k), \cdots, x_2(k-L), -x_1(k), \cdots, -x_1(k-L))^t$$

are channel outputs, and $r(t)$ is the pulse function defined by

$$r(t) = \begin{cases} 1 & \text{for } 0 \le t < T \\ 0 & \text{elsewhere} \end{cases} \tag{4}$$

where T is the symbol duration of the transmitting signals. The input-output relation of the neural network is given by

$$y = \mathbf{p}^t \mathbf{x} \tag{5}$$

The learning algorithm to update the weight is as the following:

$$\dot{\mathbf{p}} = b\{y^2 \mathbf{p} - y\mathbf{x} + \beta \mathbf{x}^t \mathbf{x}(1 - \mathbf{p}^t \mathbf{p})\mathbf{p}\} \tag{6}$$

where $b > 0$ is the constant learning rate, β is a positive number greater than one. The purpose for the first term of the learning algorithm is to increase the projection of **p** on the orthogonal subspace of input **x**, and the purpose for the second term is to decrease the projection of $\mathbf{p}$ on the input $\mathbf{x}$, the third term is a constraint to make the magnitude of $\mathbf{p}$ approach to one.

The properties of the above temporal neural network has been studied in [13]. Its results can be stated as following:

1. *Property 1*: If the input of our temporal neural network $\{\mathbf{x}_k\}$ spans a subspace $\mathcal{X} \in C^N$ with dimension $N < 2L+2$ and denote $\mathcal{X}^\perp$ the orthogonal complement of $\mathcal{X}$, then those points
$$\mathbf{p} \in \mathcal{X}^\perp \qquad \text{with } ||\mathbf{p}|| = 1$$
are the only stable equilibrium points of the neural network.
2. *Property 2*: If the average power of the input of our temporal neural network is greater than zero, i.e.,
$$\sum_{k=0}^{\infty} ||\mathbf{x}_k||^2 = \infty$$
then $||\mathbf{p}(t)|| \to 1$ as $t \to \infty$ monotonically for all initial $\mathbf{p}(0) \neq 0$.
3. *Property 3*: If for all K, $\{\mathbf{x}_k | k \geq K\}$ span the same subspace $\mathcal{X}$ with dimension less than $2L+2$, then $\mathbf{p}(t) \to \mathbf{p}_e \in \mathcal{X}^\perp$ for all initial conditions $\mathbf{p}(0) \notin \mathcal{X}$ monotonically, where $\mathbf{p}_e$ is some point in $\mathcal{X}^\perp$.

Since the input of the neural network $\mathbf{x}_k$ is the a combination of the two channel outputs, it can be expressed as following:

$$\begin{aligned} \mathbf{x}_k &= \begin{bmatrix} h_2(0) & h_2(1) & \dots & h_2(L) & 0 \\ & \ddots & & \ddots & \\ 0 & h_2(0) & h_2(1) & \dots & h_2(L) \\ -h_1(0) & -h_1(1) & \dots & -h_1(L) & 0 \\ & \ddots & & \ddots & \\ 0 & -h_1(0) & -h_1(1) & \dots & -h_1(L) \end{bmatrix} \begin{bmatrix} s(k) \\ s(k-1) \\ \vdots \\ s(k-2L-1) \end{bmatrix} \\ &= \mathbf{H}\mathbf{s}_k \end{aligned} \tag{7}$$

where $h_1(\cdot)$ and $h_2(\cdot)$ are the discrete-time impulse response of the two channels, and $s(\cdot)$ is the common input signal of the two channels.

Now we give our theorem of blind channel identification by the neural network.

Theorem: Let $\mathbf{x}(t)$ be the input of our neural network. Then $\mathbf{p}(t)$ converges to $\mathbf{h}$ up to a scalar for all initial conditions $\mathbf{p}(0) \not\perp \mathbf{h}$, if the following two conditions are satisfied:

1. The two channel responses $\hat{h}_1(z)$ and $\hat{h}_2(z)$ have no common zeros.
2. For all K, the vector constructed by the channel input sequence $\{\mathbf{s}_k | k > K\} \in R^{2L+1}$ span the entire space of R^{2L+1}.

The outline of the proof is given in Appendix.

This theorem is of particular interest because, for the first time, a neural network is presented, which is globally convergent to the channel coefficients for almost all initial conditions. The latter is true because the probability that $\mathbf{p}(0) \in \mathcal{X}$ is zero if $\mathbf{p}(0)$ is randomly chosen.

Note also that Conditions 2 is the only conditions required for $\{\mathbf{s}_k\}$. This condition does not require any statistical information and they can be satisfied by intermittent inputs.

The learning algorithm given in Eq. (6) is in the continuous form. For computation simplicity, we give the discrete counterpart of our learning algorithm with maximal learning rate as following:

$$\tilde{\mathbf{p}}(k+1) = \mathbf{p}(k) - \frac{y(k)}{||\mathbf{x}(k)||^2}\mathbf{x}(k) \tag{8}$$

$$\mathbf{p}(k+1) = \frac{\tilde{\mathbf{p}}(k+1)}{||\tilde{\mathbf{p}}(k+1)||} \tag{9}$$

where $y(k) = \mathbf{p}^t(k)\mathbf{x}(k)$ is the output of the neural network at time k, and $||\cdot||$ used in this paper is 2-norm. The proof of convergence will be shown elsewhere.

IV. Simulation

In this section, three simulations are given. In all three simulations, the inputs $\mathbf{x}(t)$ of the neural network are the outputs of two channels driven by randomly distributed 4-ary PAM signals $s(k)$. The discrete form learning is used in all three simulations.

Simulation 1: The purpose of this simulation is to show the speed, the accuracy, and the globally convergence of the neural network.

In this simulation, the actual channel impulse responses are

$$\mathbf{h}_1 = [1, 1.5, 0.5, -0.4]^t$$
$$\mathbf{h}_2 = [1, 0.9, 0.6, 0.4]^t$$

Therefore, the normalized $\mathbf{h}$ is

$$\tilde{\mathbf{h}} = [0.41, 0.61, 0.20, -0.16, 0.41, 0.37, 0.25, 0.16]^t$$

The initial condition of $\mathbf{p}$ is arbitrarily chosen to be

$$\mathbf{p}(0) = [0, 0, 0, 0, 10, 0, 0, 0]^t$$

The correlation between $\mathbf{p}(t)$ and $\tilde{\mathbf{h}}$ is shown in Figure 5, which should be one if $\mathbf{p}(t) \rightarrow \tilde{\mathbf{h}}$. The dynamics of $\mathbf{p}(t)$ is shown in Figure 6. For the clarity of illustration, only the first four components are shown, which should approach to the first four components of $\tilde{\mathbf{h}}$.

As shown in Figure 5, the correlation between $\mathbf{p}(t)$ and $\tilde{\mathbf{h}}$ approaches to 1 after only 400 symbols. This index gives the overall performance of the neural network.

Simulation 2: The purpose of this simulation is to show its ability of tracking time-varying channels.

This simulation is important because the channel under mobile telephony is usually continuously varying. The tracking ability to such time-varying channels depends on

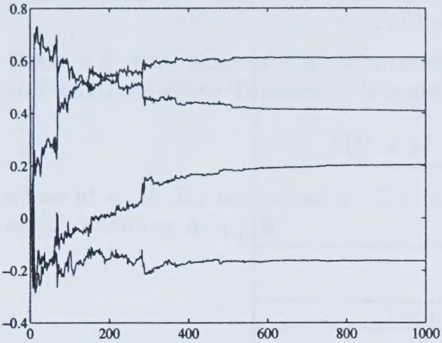


Figure 5: The dynamics of the first four components of $\mathbf{p}(t)$.

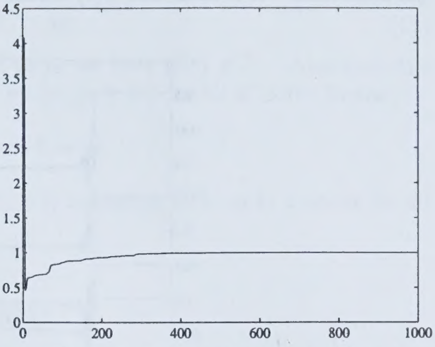


Figure 6: The correlation of $\mathbf{p}(t)$ and $\tilde{\mathbf{h}}$ vs number of symbols.

(1) the globally convergence of the algorithm, and (2) the speed of convergent rate which should be faster than the rate of changing of the channels. Since our algorithm is proven to be globally convergent, we need only to demonstrate the speed of our algorithm.

The speed of channel varying in this simulation is set at a symbol rate $300KHz$ with channel Doppler frequency $100Hz$. This speed was used in a simulator for mobile radio channels [9]. The variation of such coefficients of the first channel is plotted by the dotted lines in Figure 3, and their estimation are given by the solid lines. As it is shown, the estimated channel response can track the actual channel very well.

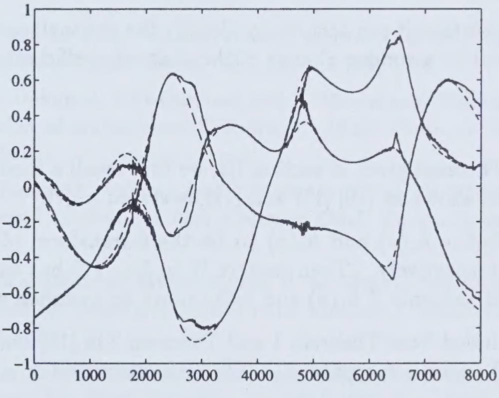


Figure 3: The trajectories of the first channel coefficients and their estimation.

Simulation 3: The purpose of this simulation is to show its robustness after the channels experienced a sudden change.

In this simulation, we simulate the case when the two channel coefficients $\tilde{\mathbf{h}}$ experienced a sudden change at the time $t = 200T$ from $[0.50, 0.25, 0.35, 0.20, 0.50, 0.40, 0.25, -0.20]^t$ to $[0.39, 0.20, 0.27, 0.16, 0.39, 0.70, 0.20, -0.16]^t$. The variation of such coefficients of the first channel is plotted by the dotted lines in Figure 4, and their estimation are given by the solid lines. It is seen that $\mathbf{p}(t)$ tracks very quickly to the sudden change of $\tilde{\mathbf{h}}$.

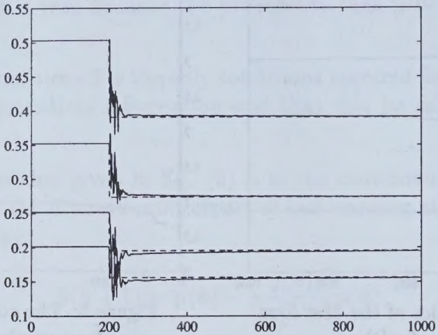


Figure 4: The trajectories of the estimates of the first channel $\mathbf{p}(t)$ following the sudden change of $\tilde{\mathbf{h}}$.

V. Conclusion

A temporal neural network for blind channel identification is presented, which is especially useful under mobile communication environment. The neural network is on-line and needs little memory.

It is shown in the Theorem that the neural network is globally convergent to the actual channel coefficients for almost all initial positions. The signal needs not to be stationary and can be intermittent in time.

Simulation shows that it can accurately identify the channel, can track time-varying channels and is robust to a sudden change of the channel coefficients.

Appendix

To prove the Theorem given in section III, we first recall a theorem about the rank property of matrix $\mathbf{H}$ shown in [10] [11] and [12] as stated in [7]:

Theorem H: Define $\hat{h}_1(z)$ and $\hat{h}_2(z)$ to be the z-transform of channel coefficient vectors $\mathbf{h}_1$ and $\mathbf{h}_2$, respectively. Then matrix $\mathbf{H}$ in Eq. (7) has full column rank, i.e. $\text{rank}(\mathbf{H}) = 2L + 1$, if and only if $\hat{h}_1(z)$ and $\hat{h}_2(z)$ share no common zeros.

It can be concluded from Theorem 1 and Theorem 2 in [13] that if the input of the neural network $\{\mathbf{x}_k\}$ spans a subspace $\mathcal{X} \in R^N$ with dimension $N < 2L + 2$, then those points

$$\mathbf{p}^* \in \mathcal{X}^\perp \qquad \text{with } ||\mathbf{p}^*|| = 1$$

are the only stable equilibrium points of the neural network. Furthermore, $\mathbf{p}(t) \rightarrow \mathbf{p}^* \in \mathcal{X}^\perp$ with $||\mathbf{p}^*|| = 1$ as $t \rightarrow \infty$ for all initial conditions $\mathbf{p}(0) \notin \mathcal{X}$ if $\mathcal{X}$ is spanned by $\{\mathbf{x}_k | k \geq K\}$ for all $K > 0$.

Conditions 1 and 2 of our Theorem in section III imply that the dimension of subspace $\mathcal{X}$ spanned by $\{\mathbf{x}_k \in R^{2L+2}\}$ is $2L + 1$, and the dimension of $\mathcal{X}^\perp$ is one. Because

$$\mathbf{H}^t\mathbf{h} = 0 \tag{10}$$

we have

$$\mathbf{x}^t(t)\mathbf{h} = 0 \qquad \forall t \tag{11}$$

Hence $\mathbf{h} \in \mathcal{X}^{\perp}$. Since $\mathbf{p}(0)$ is not orthogonal to $\mathbf{h}$, we have $\mathbf{p}(0) \notin \mathcal{X}$. Also from the two conditions of our Theorem, $\mathcal{X}$ is spanned by $\{\mathbf{x}_k | k \geq K\}$ for all $K > 0$. Hence

$$\mathbf{p}(t) \rightarrow \mathbf{p}^* \qquad \text{as } t \rightarrow \infty$$

where $\mathbf{p}^* = \pm\tilde{\mathbf{h}}$, the normalized $\mathbf{h}$. Therefore, $\mathbf{p}(t)$ converges to $\mathbf{h}$ up to a scalar for all initial conditions $\mathbf{p}(0) \not\perp \mathbf{h}$.

REFERENCES

[1] Y. Sato, "A method of self-recovering equalization for multilevel amplitude-modulation," *IEEE trans. on Commum.*, pp. 679-682, June 1975.

[2] A. Benveniste, M. Goursat, and G. Ruget, "Robust identification of a nonminimum phase system: blind adjustment of a linear equalizer in data communications.," *IEEE trans. on Automatic Control*, pp. 385-399, June 1980.

[3] G. Giannakis, Y. Inouye, and J. Mendel, "Cumulant-based identification of multichannel moving average models," *IEEE Trans. on Automat. Contr.*, vol. 34, pp. 783-787, 1989.

[4] O. Shalvi and E. Weinstein, "New criteria for blind deconvolution of nonminimum phase systems (channels)," *IEEE Trans. on Inform. Theory*, pp. 312-320, March 1990.

[5] L. Tong, G. Xu, and T.Kailath, "A new approach to blind identification and equalization of multipath channels," in *Proc. 25th Asilomar Conf.*, (Pacific Grove, CA), pp. 340-349, 1991.

[6] H. Liu, G. Xu, and L. Tong, "A deterministic approach to blind equalization," in *Proc. 27th Asilomar Conf.*, (Pacific Grove, CA), pp. 751-755, 1993.

[7] E. Moulines, P. Duhamel, J-F. Cardoso, and S. Mayrargue, "Subspace methods for the blind identification of multichannel FIR filters," *IEEE Trans. on Signal processing,* pp. 516-525, Feb. 1995.

[8] B. Khalaj, B. Hassibi, A. Paulraj, and T. Kailath, "Blind identification of FIR channels via antenna array," in *Proc. of 27th Asilomar Conf.*, Pacific Grove, CA, pp. 721-725, 1993.

[9] S. Chen, S. McLaughlin, B. Mulgrew, and P. M. Grant, "Adaptive Bayesian decision feedback equalizer for dispersive mobile radio channels," *IEEE Trans. Commun.*, pp. 1937-1946, May 1995.

[10] L. Tong, G. Xu, and T. Kailath, "Blind identification and equalization of multipath channels," in *Proc. Int. Conf. Commun.*, pp. 1513-1517, June 1992.

[11] Y. Li and Z. Ding, "Blind channel identification based on second order cyclostationary statistics," in *Proc. ICASSP*, vol. 4, pp. 81-84, 1993

[12] L. Tong, G. Xu, and T. Kailath, "Fast blind equalization via antenna arrays," in *Proc. ICASSP*, vol. 4, pp. 272-275, 1993.

[13] G. Dong and R. Liu, "A globally convergent neural network for blind channel identification," Technical Report: TR-95-05-01, University of Notre Dame. Also, submitted to publish.

Information Theoretic Lower Bounds on Convergence Rates of Nonparametric Estimators

Andrew Barron and Yuhong Yang
Department of Statistics
Yale University
P.O. Box 208290, Yale Station
New Haven, CT 06520

Abstract

Bounds on minimax rates of convergence are derived based on certain information-theoretic considerations. These bounds depend only on metric entropy conditions and are used to identify the minimax rates of convergence for nonparametric estimations.

I. INTRODUCTION

In this paper, we present some general results on determining lower bounds on statistical risks and the minimax rates of convergence for density estimation.

In statistical decision theory, the minimax method is of importance. The minimax risk provides us with the insight into the limitation we can not overcome for the worst case. Knowing the minimax rates of convergence, we can evaluate various estimators by comparing their risks with the minimax ones.

The minimax rates of convergence are often determined in two steps. Firstly, a good lower bound is obtained for the target family of densities. Then a specific estimator is constructed so that the maximum risk is within a constant factor of the derived lower bound, which enables us to conclude about the minimax rate of convergence. For nonparametric function estimations, two methods are often used to derive the minimax lower bound: Fano's inequality and Assouad's Lemma. The first one is used in Hasminskii (1978), Ibragimov and Hasminskii (1980, 1982 and 1990), Efroimovich and Pinsker (1982), Birge (1983). The second one is utilized in Bretagnolle and Huber (1979), Stone (1980), Assouad (1983), and Birge (1986).

Birge (1986) claims that Fano's inequality is more general and could replace Assouad's Lemma in almost all practical situations. Yu (1995) gives a lower bound similar to Assouad's in terms of Kullback-Leibler distance using Fano's inequality. However, for the derivation of lower bounds, in Yu (1995), Birge (1986) and Devroye (1987) and some other papers, Fano's inequality is used in a weak way, where a very rough upper bound is applied for the mutual information between the parameter and the observations. In this paper, by applying a sharper bound on the mutual information based on some more information-theoretic considerations, we show that the minimax lower bound could be easily obtained directly from Fano's inequality for general calsses of densities satisfying a certain metric entropy type condition. On the contrary, without applying Fano's inequality in the better way, to obtain the right order lower bound, Assouad's Lemma was used to a special subclass produced from perturbations of the densities (see Birge (1986)), or more efforts were put in the construction of finite subsets of densities on which Fano's inequality was applied (see Hasminskii (1978), Ibragimov and Hasminskii (1980 and 1982)). These constructions seem to be much more complicated and they rely on more properties of the target density class than the entropy structure.

II. MAIN RESULTS

Suppose we have a class of densities $\{p_\theta : \theta \in S\}$. The parameter space S could be a finite dimensional space or a nonparametric space (e.g., the density class itself). Let $X_1, X_2, ..., X_n$ be an i.i.d. sample from $p_\theta, \theta \in S$. We want to estimate the true density p_θ. Both the relative entropy loss and the Hellinger loss will be considered in this paper.

Our results are based on certain metric entropy conditions. We first give a definition.

Let d be a mapping from $S \times S$ to R^+. Such a mapping is called a distance on S.

Definition: A finite set $S_0 \subset S$ is said to be an ϵ-packing set for $\epsilon > 0$ if for any $\theta, \theta' \in S_0, \theta \neq \theta'$, we have $d(\theta, \theta') \geq \epsilon$, and for any $\tilde{\theta} \in S$, there exists a $\theta_0 \in S_0$ such that $d(\tilde{\theta}, \theta_0) \leq \epsilon$. We call ϵ the packing radius.

In the above definition, we do not require the distance d to be a metric. In fact, one choice of d will be the relative entropy distance. Let $d_K^2(\theta, \theta') = D(p_\theta \parallel p_{\theta'})$. Clearly $d_K(\theta, \theta')$ is asymmetric in its two arguments, so it can not be a metric. The major shortcoming of this distance is that there is no triangle-like inequalities, that is, there might not exist two constants c and C such that $d_K(\theta, \theta') \geq C d_K(\theta, \theta'') - c d_K(\theta'', \theta')$ for any θ, θ' and θ'' in S. The inequality is usually used to rule out the possibility that one density estimator is too close in d to too many densities which are far away from each other in the same distance d. This might happen with the relative entropy distance. It seems necessary to have some additional conditions enabling a triangle-like inequality to obtain a reasonable lower bound in terms of d-distance.

Another distance involved is the Hellinger distance $d_H(\theta, \theta') = \sqrt{\int \left(\sqrt{p_\theta} - \sqrt{p_{\theta'}}\right)^2 d\mu}$. It is well known that Hellinger distance is a metric.

Let $M_K(\epsilon)$, $M_H(\epsilon)$ be the largest log-cardinality of an ϵ-packing set under d_K and d_H respectively. Assume $M_K(\epsilon)$ and $M_H(\epsilon)$ are finite for each $\epsilon > 0$ and they approach to ∞ as $\epsilon \to 0$.

1. Minimax lower bound.

Choose ϵ_n such that $M_K(\epsilon_n) = n\epsilon_n^2$. Let $\underline{\epsilon}_n$ be determined by

$$M_H(\epsilon) = 2(2n\epsilon_n^2 + \log 2)\}$$

and let $F_{\underline{\epsilon}_n}$ be the $\underline{\epsilon}_n$-packing set in S under d_H. Let G_{ϵ_n} be an ϵ_n-packing set in S under d_K. By definition, $\log |G_{\epsilon_n}| \leq M_K(\epsilon_n)$.

For the relative entropy risk,

$$\min_{\hat{p}} \max_{\theta \in S} E_\theta D(p_\theta \quad \| \quad \hat{p}) \geq \min_{\hat{p}} \max_{\theta \in F_{\underline{\epsilon}_n}} E_\theta D(p_\theta \parallel \hat{p})$$

$$\begin{aligned} &\geq \min_{\hat{p}} \max_{\theta\in F_{\underline{\epsilon}_n}} E_\theta d_H^2(p_\theta,\hat{p}) \\ &\geq \min_{\hat{p}} \max_{\theta\in F_{\underline{\epsilon}_n}} \left(\frac{\underline{\epsilon}_n}{2}\right)^2 E_\theta I_{\{d_H(p_\theta,\hat{p})\geq \frac{\underline{\epsilon}_n}{2}\}} \,. \end{aligned}$$

Define $\tilde{p} = p_{\tilde{\theta}}$ if $\tilde{\theta} = \min_{\theta'} d_H^2(p_{\theta'},\hat{p})$. Then

$$\begin{aligned} \min_{\hat{p}} \max_{\theta\in S} E_\theta D(p_\theta \parallel \hat{p}) &\geq \min_{\hat{p}} \max_{\theta\in F_{\underline{\epsilon}_n}} \left(\frac{\underline{\epsilon}_n}{2}\right)^2 P_\theta\left(\tilde{\theta}\neq\theta\right) \\ &\geq \min_{\hat{p}} \left(\frac{\underline{\epsilon}_n}{2}\right)^2 \sum_{\theta\in F_{\underline{\epsilon}_n}} w(\theta) P_\theta\left(\tilde{\theta}\neq\theta\right) \\ &= \min_{\hat{p}} \left(\frac{\underline{\epsilon}_n}{2}\right)^2 P_w\left(\tilde{\theta}\neq\theta\right), \end{aligned}$$

where w is any prior on $F_{\underline{\epsilon}_n}$. By Fano's inequality, taking w_0 as the uniform prior on $F_{\underline{\epsilon}_n}$,

$$P_{w_0}\left(\tilde{\theta}\neq\theta\right) \geq 1 - \frac{I\left(\Theta_0; X^n\right)+\log 2}{\log|F_{\underline{\epsilon}_n}|},$$

where $\Theta_0 = F_{\underline{\epsilon}_n}$. But

$$\begin{aligned} I\left(\Theta_0;X^n\right) &= \sum_{\theta\in F_{\underline{\epsilon}_n}} w_0(\theta)p(x^n|\theta)\log\frac{w_0(\theta)p(x^n|\theta)}{w_0(\theta)\sum_\theta w_0(\theta)p(x^n|\theta)}dx^n \\ &= \sum_{\theta\in F_{\underline{\epsilon}_n}} w_0(\theta)p(x^n|\theta)\log\frac{p(x^n|\theta)}{\sum_\theta w_0(\theta)p(x^n|\theta)d\theta}dx^n \\ &\leq \sum_{\theta\in F_{\underline{\epsilon}_n}} w_0(\theta)p(x^n|\theta)\log\frac{p(x^n|\theta)}{p_{w_1}(x^n)}dx^n \\ &\leq \max_{\theta\in F_{\underline{\epsilon}_n}} D\left(P_{X^n|\theta} \parallel P_{w_1,X^n}\right), \end{aligned}$$

where P_{w_1,X^n} has the mixture density $\sum_{\theta\in S} w_1(\theta)p(x^n|\theta)$ with respect to any given prior w_1 on S. The first inequality above follows from the fact that w_0 minimizes

$$\sum_{\theta\in F_{\underline{\epsilon}_n}} w_0(\theta)p(x^n|\theta)\log\frac{p(x^n|\theta)}{q(x^n)}dx^n$$

over $q(x^n)$ being any density. Choose w_1 to be the uniform prior on G_{ϵ_n}. Then because G_{ϵ_n} is an ϵ_n- packing in S under d_K, for each $\theta\in S$, there exists $\tilde{\theta}\in G_{\epsilon_n}$ such that $D(p_\theta \parallel p_{\tilde{\theta}}) = d_K^2(\theta,\tilde{\theta}) \leq \epsilon_n^2$. It follows that

$$\begin{aligned} D\left(P_{X^n|\theta} \parallel P_{w_1,X^n}\right) &= E\log\frac{p(X^n|\theta)}{\frac{1}{|G_{\epsilon_n}|}\sum_{\theta'}p(X^n|\theta')} \\ &\leq \log|G_{\epsilon_n}| + D\left(P_{X^n|\theta} \parallel P_{X^n|\theta'}\right) \\ &\leq M_K(\epsilon_n) + n\epsilon_n^2. \end{aligned}$$

By the choice of $\underline{\epsilon}_n$,

$$\frac{I\left(\Theta_0;X^n\right)+\log 2}{\log|F_{\underline{\epsilon}_n}|} \leq \frac{1}{2}.$$

Therefore

$$\min_{\hat{p}} \max_{\theta \in S} E_\theta D(p_\theta \parallel \hat{p}) \geq \frac{\underline{\epsilon}_n^2}{8}.$$

Remark: In the above analysis, if the Hellinger distance is replaced by another metric d_1 on S, then we have a similar lower bound on the minimax risk under d_1 loss $\min_{\hat{\theta}} \max_{\theta \in S} d_1(\theta, \hat{\theta}) \geq \frac{\overline{\underline{\epsilon}}_n^2}{8}$, where $\overline{\underline{\epsilon}}_n$ satisfies $M_{d_1}(\epsilon) = 2(2n\epsilon_n^2 + \log 2)$ with M_{d_1} being the packing number under d_1.

2. Minimax upper bound.

To provide an upper bound on the minimax rate of convergence, we construct an estimator as follows. Consider the ϵ_n-packing set G_{ϵ_n} in S under d_K and the uniform prior w_1 on G_{ϵ_n}. Let

$$\overline{p}(x) = \frac{1}{n} \sum_{i=0}^{n-1} \widehat{p}_i, \qquad \text{where } \widehat{p}_i = p\left(x_{i+1} = x | X^i\right).$$

Then by convexity and chain rule,

$$\begin{aligned} ED(p_\theta \quad \parallel \quad \overline{p}) &\leq \frac{1}{n} \sum_{i=0}^{n-1} ED(p_\theta \parallel \widehat{p}_i) \\ &= \quad D\left(P_{X^n|\theta} \parallel p_{w_1, X^n}\right) \leq \frac{1}{n}\left(M_K(\epsilon_n) + n\epsilon_n^2\right) = 2\epsilon_n^2. \end{aligned}$$

Thus

$$\min_{\hat{p}} \max_{\theta \in S} E_\theta D(p_\theta \parallel \hat{p}) \leq 2\epsilon_n^2.$$

From the above lower and upper bounds, we have the following theorem on the minimax relative entropy risk.

Theorem: For the relative entropy risk, we have

$$\frac{\underline{\epsilon}_n^2}{8} \leq \min_{\hat{p}} \max_{\theta \in S} E_\theta D(p_\theta \parallel \hat{p}) \leq 2\epsilon_n^2.$$

If $\frac{\underline{\epsilon}_n^2}{8}$ and $2\epsilon_n^2$ converge to 0 at the same rate, then the minimax rate of convergence is identified. For $\underline{\epsilon}_n^2$ and ϵ_n^2 to be of the same order, it is sufficient that the following two conditions hold: (1). there exist two positive constants a and b such that $\sup\{\epsilon : M_K(\epsilon) \geq (1+a)n\epsilon_n^2\} \geq b\epsilon_n$, and (2). $M_H(\epsilon)$ is of the same order as $M_K(\epsilon)$ for ϵ being of the same rate as ϵ_n.

The second requirement above is the equivalence of the entropy structure under the relative entropy distance and that under Hellinger distance when ϵ is small. Since Hellinger distance is always bounded above by the relative entropy, a simple sufficient (but not necessary) condition to ensure the second requirement is that the density in the target class is uniformly bounded above and away from 0. The first requirement is satisfied if

$$\underline{\lim}_{\epsilon \to 0} \frac{M_K(\frac{\epsilon}{2})}{M_K(\epsilon)} > 1.$$

The above conditions are satisfied for the usual smooth nonparametric classes defined in terms of derivatives. Then we have $\underline{\epsilon}_n \asymp \epsilon_n$. For instance, for the densities whose logarithms are in the Sobolev balls, the above two requirements are both satisfied. Thus the minimax rate of convergence is determined. However, for not very rich classes of densities, the lower bound and the upper bound derived in the above way do not converge at the same rate. For instance, for a finite-dimensional class, both $M_K(\epsilon)$ and $M_H(\epsilon)$ might be of order $\log\left(\frac{1}{\epsilon}\right)^m$ for some constant $m \geq 1$. Then ϵ_n and $\underline{\epsilon}_n$ are not of the same order with $\epsilon_n \asymp \frac{\sqrt{\log n}}{\sqrt{n}}$ and $\underline{\epsilon}_n = o(\frac{1}{\sqrt{n}})$. Thus the upper bound provided by the theorem is only suboptimal and the lower bound is too small. For smooth finite-dimensional models, the minimax risk can be solved using some traditional statistical methods such as Bayes procedures, C-R inequality, Van Tree's inequality, etc. But these techniques require more than the entropy condition. In a more recent paper, we use localized conditions to derive minimax bounds which are suitable for both parametric and nonparametric classes of densities.

References:

1. Assouad, P.: Deux remarques sur l'estimation. C.R. Acad. Sci. Paris 296, Ser.I, 1021-1024 (1983)
2. Birge, L.: On estimating a density using Hellinger distance and some other strange facts. Probab. Th. Rel. Fields, 71, 271-291 (1986)
3. Birge, L.: Approximation dans les espaces metriques et theorie de l'estimation. *Z. Wahrscheinlichkeitstheor. Verw. Geb.* 65, 181-237 (1983)
4. Devroye, L.: A course in density estimation. Boston: Birkauser (1987)
5. Bretagnolle, J., Huber, C.: Estimation des densites: risque minimax. Z. Wahrscheinlichkeitstheor. Verw. Geb. 47, 119-137 (1979)
6. Efroimovich, S. Yu., Pinsker, M.S.: Estimation of square-integrable probability density of a random variable. *Problemy Peredachi Informatsii* 18 (3), 19-38 (1982)
7. Hasminskii, R.Z.: A lower bound on the risks of nonparametric estimates of densities in the uniform metric. *Theory probab. Appl.* 23, 794-796 (1978)
8. Ibragimov, I.A., Hasminskii, R.Z.: Estimation of distribution density. Zap. Nauchn. Semin. LOMI 98, 61-85 (1980)
9. Ibragimov, I.A., Hasminskii, R.Z.: Bounds for the risks of non-parametric regression estimates. Theory probab. Appl. 27, 84-99 (1982)
10. Stone, C.J.: Optimal rates of convergence for nonparametric estimators. Ann. Statist. 8, 1348-1360 (1980)
11. Yu, B.: Assouad, Fano, and Le Cam. To appear in *Festschrift in honor of L. Le Cam on his 70th birthday* (1995)

Some Results on Nonparametric Regression Estimation For Arbitrary Random Processes*

S.E. Posner[†]
Department of Statistics
University of Toronto
Toronto, ONT M5S 1A1
posner@utstat.toronto.edu

S.R. Kulkarni
Department of Electrical Engineering
Princeton University
Princeton, NJ 08544
kulkarni@ee.princeton.edu

Abstract

Let $X_1, X_2, \ldots$ be an arbitrary random process taking values in a compact subset of a separable metric space. Each X_i has an associated Y_i drawn from an unknown conditional distribution $F(y|X_i = x)$. We study nonparametric regression estimates of the form $\sum_{i=1}^{n-1} W_{ni}(X_1, \ldots, X_n)Y_i$ for estimating Y_n based on X_n and the data $\{(X_i, Y_i)\}_{i=1}^{n-1}$. Our work extends previous analysis of such estimators to the case where $\{X_i\}$ is a completely arbitrary random process. We introduce a notion of Cesáro consistency for which the time average of the estimation errors is required to converge to zero. Necessary and sufficient conditions on weight sequences $\{W_{ni}\}$ for Cesáro consistency are established. Certain special cases are obtained, such as nearest neighbor, kernel, and partitioning estimators along with their respective rates of convergence. Our technique is an elementary sample path analysis which leads to very general results. One consequence of our work is a recovery and extension of some classical results. In particular, we make new statements about stationary processes and in general separable metric space settings.

1 Introduction

Let $X_1, X_2, \ldots, X_n$ be an arbitrary random process taking values in a subset of a general separable metric space $(\mathcal{X}, \rho)$. Special cases include non-stationary or non-ergodic processes and deterministic sequences. Each $X_i = x_i$ has an associated label Y_i which is a random variable drawn from an unknown conditional distribution $F(y|X_i = x_i)$ taking values in a Hilbert space $\mathcal{Y}$. A classical problem that has received much attention is to nonparametrically estimate Y_n given X_n. One body of work focuses on the situation in which the distribution of (X_n, Y_n) is unknown but the estimator uses additional data pairs $\{(X_i, Y_i)\}_{i=1}^{n-1}$ to estimate the regression function $m(X_n) = E[Y_n|X_n]$. Following the seminal work of Stone (1977), we focus our attention to estimators of the form

$$m_n(X_n) = \sum_{i=1}^{n-1} W_{ni}(X_1, \ldots, X_n)Y_i, \tag{1}$$

where $\{W_{ni}\}_{i=1}^{n-1}$ is a sequence of weights that satisfy certain conditions. This estimate is a function of all the observed data.

---

*This work was supported in part by the National Science Foundation under grant IRI-9457645 and by the U.S. Army Research Office under grant DAAL03-92-G-0320.

[†] This work was done while the author was with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544.

Most previous work has considered the case in which the data pairs $\{(X_i, Y_i)\}$ are independent and identically distributed (i.i.d.), although some recent work considered the stationary case. See, for example, Györfi, Härdle, Sarda, Vieu (1989), Truong and Stone (1992). In the i.i.d. case, various notions of convergence of the estimate have been defined and corresponding conditions for convergence have been obtained. Following Györfi (1981b) and others, the estimate $m_n(X)$ of $m(X_n)$ is said to be *weakly consistent* if

$$\lim_{n\to\infty} E|m_n(X_n) - m(X_n)|^p = 0, \ \forall p \geq 1.$$

The estimate is *strongly consistent* if

$$\lim_{n\to\infty} m_n(X_n) = m(X_n) \ \text{ almost surely (a.s.)}$$

The estimate is strongly (weakly) *universally consistent* if it is strongly (weakly) consistent for all reasonable distributions $F(x, y)$, i.e., it is necessary to assume either a finite moment or uniform bound on Y. Stone (1977) proved necessary and sufficient conditions on the weights for weak consistency without making any assumptions on the distributions other than that the $\{(X_i, Y_i)\}$ are i.i.d. and that $E|Y|^p < \infty$ for all $p \geq 1$. Subsequent papers proved universal consistency for certain special cases of (1). Consistency of the k_n-NN rule was shown by Devroye (1981, 1982), Györfi (1981b), and Stone (1977). Consistency for the kernel estimator was shown by Devroye (1981), Devroye and Wagner (1980), Györfi, Krzyżak, and Pawlak (1984), Györfi (1981b), Krzyżak, and Pawlak (1984). Consistency of partitioning estimators was established by Breiman, Friedman, Olshen, and Stone (1984), Devroye and Györfi (1983), Gordon and Olshen (1980, 1984), and Györfi (1991). To obtain rates of convergence requires additional assumptions on the distribution. Results by Györfi (1981a), Krzyżak (1986), and Stone (1980, 1982) provide rates of convergence in a variety of scenarios.

In this paper, we impose no restrictions on the random process $\{X_i\}$ except that it take values in a compact subset of a separable metric space and that given X_i, the label Y_i is conditionally independent of $\{(X_j, Y_j)\}_{i \neq j}$. This is a very general setting and as a result, in general, one cannot expect to design a consistent estimator. However, we will show that when restricted to continuous regression functions, we can prove that conditions on the weights analogous to those in Stone (1977) ensure time-average or Cesáro consistency, i.e., the time-averaged estimation errors go to zero for every random process. Our proof techniques are rather elementary sample path analyses and are in line with current trends in information theory and statistics to evaluate performance in terms of individual data sequences. See, for example, Cover (1991), Feder, Merhav, and Gutman (1992), Kulkarni and Posner (1994).

In Section 2 we discuss some preliminaries. Section 3 contains the main theorem of the paper which gives necessary and sufficient conditions for Cesáro consistency. In Section 4 we give results that traditional consistent estimators such as the nearest neighbor, kernel, and partitioning estimators are in fact also universally Cesáro consistent. We also provide rates of convergence for these estimators under additional smoothness conditions. In Section 5, we partially recover and extend Stone's result and provide some new results. for separable metric spaces and stationary processes in the case of continuous regression functions.

2 Preliminaries

Consider a sequential estimation problem as follows. Let $X_1, X_2, \ldots, X_n$ be an arbitrary random process taking values in a subset A of a separable metric space $(\mathcal{X}, \rho)$ and let $Y_1, Y_2, \ldots, Y_n$ be a corresponding sequence of random variables taking values in a Hilbert space $\mathcal{Y}$ equipped with an inner product induced norm $|\cdot|$. For concreteness, one can consider $\mathcal{Y} = \mathbf{R}$ with the usual Euclidean norm $|\cdot|$. Throughout the paper, the norm $|\cdot|$ is meant to be understood from the context. We impose the following assumption on the pairs (X_i, Y_i) which implies that given X_i, the label Y_i is conditionally independent of $\{(X_j, Y_j)\}_{j \neq i}$ and drawn according to $F(y|X_i)$.

(A0)

$$Pr(Y_i \in S | X_1, \ldots, X_n, Y_1, \ldots, Y_{i-1}, Y_{i+1}, Y_n)$$

$$= Pr(Y_i \in S | X_i) = \int_S F(dy|X_i)$$

for each i for every measureable set S.

Our goal is to sequentially estimate $m(X_n) = E[Y_n|X_n]$ using only the data $\mathcal{D}_n = X_n \cup \{(X_i, Y_i)\}_{i=1}^{n-1}$. This problem formulation is quite general in that we do not make any assumptions on the sequence $\Omega_n = X_1, X_2, \ldots, X_n$ other than (A0).

We consider estimators of the form

$$m_n(X_n) = \sum_{i=1}^{n-1} W_{ni}(\Omega_n) Y_i, \tag{2}$$

where $W_{ni}(\Omega_n) = W_{ni}(X_1, X_2, \ldots, X_n), 1 \leq i < n$. Intuitively, a good estimator will weight more heavily those i's for which X_i is near X_n.

For each $p \geq 1$, the p-loss is defined as

$$r_n^{(p)}(\Omega_n) = (E[|m_n(X_n) - m(X_n)|^p | \Omega_n])^{1/p},$$

and the time-averaged p-loss is defined as

$$a_n^{(p)}(\Omega_n) = \frac{1}{n} \sum_{i=2}^{n} r_i^{(p)}(\Omega_i). \tag{3}$$

Throughout this paper we impose the following assumptions on the conditional distribution $F(y|x)$:

(A1) $\sup_{x \in A} E[|Y|^p | X = x] = M_p < \infty$, $\forall p \geq 1$,

(A2) the regression function $m(x) = E[Y|X = x]$ is a continuous function.

We define the following new notion of consistency which will be shown to imply weak consistency in certain special cases.

Definition 1 *Let $\Omega_n = X_1, X_2, \ldots, X_n$ be an arbitrary random process taking values in $A \subset (\mathcal{X}, \rho)$ with $(X_1, Y_1), \ldots, (X_n, Y_n)$ satisfying (A0). The estimate $m_n(X_n) = \sum_{i=1}^{n-1} W_{ni}(\Omega_n) Y_i$ of $m(X_n)$ is said to be* Cesáro consistent for Ω_n *if*

$$\lim_{n \to \infty} a_n^{(p)}(\Omega_n) = 0 \ a.s. \quad \forall p \geq 1, \tag{4}$$

for $F(y|x)$ that satisfies (A1) and (A2).

Note that when we say that an estimate is Cesáro consistent, we mean that (4) holds for every conditional distribution with a continuous regression function and finite moments. Discontinuous regression functions with arbitrary $\{X_i\}$ may lead to unbounded cumulative loss.

Definition 2 *The estimate $m_n(X_n) = \sum_{i=1}^{n-1} W_{ni}(\Omega_n) Y_i$ of $m(X_n)$ is said to be* universally Cesáro consistent *if it is Cesáro consistent for every random process Ω_n.*

We are also interested in rates of convergence; although for this we need to impose additional smoothness conditions. We will consider those regression functions that satisfy a Lipschitz condition. Let $Lip[K, \alpha]$, for some $K > 0$ and $0 < \alpha \leq 1$, denote the set of all functions $m(x)$ such that

$$\text{(A3)} \qquad |m(x_1) - m(x_2)| \leq K\rho(x_1, x_2)^\alpha \quad \forall x_1, x_2 \in A \subset \mathcal{X}.$$

3 Cesáro Consistent Nonparametric Regression

We will now find necessary and sufficient conditions on the weights $\{\overline{W}_n\}$ for Cesáro consistency. Our result is an extension of Stone (1977) to general metric spaces and to arbitrary random processes. To achieve the latter, we can only conclude that the time-averaged loss goes to zero. Accordingly, it is natural that our statements on the weights are time-averaged versions of those in Stone (1977). It should be noted that as we assume continuity of the regression function, we do not require the first of the five conditions in Stone (1977).

Note that $\overline{W}_n$ is called a probability weight if $W_{ni} \geq 0$ for each i and if $\sum_{i=1}^{n-1} W_{ni} = 1$. For notational convenience define $d_{ni} = \rho(X_n, X_i)$ for any $X_n, X_i \in (\mathcal{X}, \rho)$.

Theorem 1 (Cesáro Consistency) *Let $\Omega_n = \{X_n\}$ be an arbitrary random process taking values in a compact subset $A \subset (X, \rho)$ with $(X_1, Y_1), \ldots, (X_n, Y_n)$ satisfying (A0). Let $\{\overline{W}_n\}$ be a sequence of weights. If the following conditions hold:*

(C1) $\sum_{i=1}^{n-1} |W_{ni}(\Omega_n)| \leq D$ *a.s., for all* $n > 1$ *for some* $D > 0$,

(C2) $\frac{1}{N}\sum_{n=2}^{N}\sum_{i=1}^{n-1} |W_{ni}(\Omega_n)| I(d_{ni} > \epsilon) \to 0$ *a.s., for all* $\epsilon > 0$,

(C3) $\frac{1}{N}\sum_{n=2}^{N}\left|\sum_{i=1}^{n-1} W_{ni}(\Omega_n) - 1\right| \to 0$ *a.s.,*

(C4) $\frac{1}{N}\sum_{n=2}^{N}\max_i |W_{ni}(\Omega_n)| \to 0$ *a.s.,*

then the estimate $m_n(X_n) = \sum_{i=1}^{n-1} W_{ni}(\Omega_n)Y_i$ of $m(X_n)$ is Cesáro consistent for Ω_n. Conversely, if the estimate using weights $\{\overline{W}_n\}$ is Cesáro consistent for Ω_n then (C3) and (C4) hold. If, in addition, $\overline{W}_n \geq 0$ then (C2) holds.

Hence, if $\{\overline{W}_n\}$ are probability weights, then the estimate is Cesáro consistent for Ω_n if and only if (C2) and (C4) hold.

Remarks

Recall that Cesáro consistency means that (4) holds for every conditional distribution that satisfies (A1) and (A2), and for $(X_1, Y_1), \ldots, (X_n, Y_n)$ that satisfy (A0).

There is no converse statement for (C1) since it is a pointwise condition which is not implied by Cesáro consistency.

If (C1)—(C4) hold for every random process $\{X_i\}$ then the estimate is universally Cesáro consistent. In Section 4 we will construct weights for which this holds.

Rates of convergence can also be obtained in terms of rates at which the terms in (C2) - (C4) tend to zero.

4 Universal Cesáro Consistency for Specific Estimators

We will now analyze a variety of specific weight sequences that have received much attention in the literature. The statement in Theorem 1 gives conditions on the weights that depend on the actual random process $\{X_i\}$. In the sequel, for particular $\{\overline{W}_n\}$ we find conditions on the weights that are independent of $\{X_i\}$. This gives universal Cesáro consistency. We will also provide rates of convergence for the specific weight sequences considered.

Let $\mathcal{B}_{x,\epsilon}$ denote the open ball of radius ϵ about a point $x \in \mathcal{X}$, and let $A \subset \mathcal{X}$. As in Kolmogorov and Tihomirov (1961), the *metric covering number* $\mathcal{N}(\epsilon, A) \equiv \mathcal{N}(\epsilon, A, \rho)$ is defined as the smallest number of open balls of radius ϵ that cover the set A. That is,

$$\mathcal{N}(\epsilon, A) = \inf\{k : \exists x_1, \ldots, x_k \in \mathcal{X} \quad s.t. \quad A \subset \cup_{i=1}^{k} \mathcal{B}(x_i, \epsilon)\}.$$

The *metric covering radius* $\mathcal{N}^{-1}(k, A) \equiv \mathcal{N}^{-1}(k, A, \rho)$ is defined as the smallest radius such that there exist k balls of this radius which cover the set A. That is,

$$\mathcal{N}^{-1}(k, A) = \inf\{\epsilon : \exists x_1, \ldots, x_k \in \mathcal{X} \quad s.t. \quad A \subset \cup_{i=1}^{k} \mathcal{B}_{x_i,\epsilon}\}.$$

4.1 General Nearest Neighbor Regression Estimation

Much interest in nearest neighbor (NN) regression estimation stemmed from the work of Cover and Hart (1967, 1968). The general nearest neighbor estimation procedure is to consider a linear combination of all Y_i's with the closer ones receiving larger weighting. The estimate is defined as

$$m_n(X_n) = \sum_{i=1}^{n-1} W_{ni}(\Omega_n) Y_i = \sum_{i=1}^{n-1} v_{ni} Y'_{ni},$$

where $\{Y'_{ni}\}$ is the NN ordering of $\{Y_i\}$, and $v_n = (v_{n1}, \ldots, v_{nn-1})$ is a probability vector of weights. That is, Y'_{ni} is the Y_j associated with $X_j = X'_{ni}$, where X'_{ni} is the i^{th} closest sample to X_n. Normally, the v_{ni} are chosen to give more weight to nearer samples. Let $d_n(i) = \rho(X_n, X'_{ni})$, the i^{th} NN distance. In traditional i.i.d. sampling it turns out that certain conditions on the weights $\{v_n\}$ lead to weak and strong universal consistency. Stone (1977) obtained conditions that ensure weak universal consistency. Conditions for strong universal consistency can be found in Devroye (1981, 1982) and Györfi (1981b). Rates of convergence under a Lipschitz smoothness assumption were established by Györfi (1981a), and extended to general metric spaces by Kulkarni and Posner (1994). Kulkarni and Posner (1994) also show that for an arbitrary random process we have that the time average of the NN distances goes to zero at a rate that depends only on the covering numbers of the underlying space. This led to a result for k_n-NN estimation under arbitrary sampling.

The following theorem extends the result by Kulkarni and Posner (1994) to general NN regression estimates. The first part of the theorem shows that we can construct NN weights that satisfy the conditions in Theorem 1. We then give rates of convergence of the time-averaged risks, $a_n^{(p)}$.

Theorem 2 *Let $\Omega_n = X_1, X_2, \ldots, X_n$ be an arbitrary random process taking values in a compact subset A of $(\mathcal{X}, \rho)$ with $(X_1, Y_1), \ldots, (X_n, Y_n)$ satisfying (A0). If the following conditions hold:*

(N1) there exists a sequence $\{k_n\}$ such that as $n \to \infty$,

$$k_n \to \infty, k_n/n \to 0, \text{ and } \sum_{i=k_n+1}^{n-1} v_{ni} \to 0,$$

(N2) $\max_i v_{ni} \to 0$,

then the NN estimate $m_n(X_n) = \sum_{i=1}^{n-1} v_{ni} Y'_{ni}$ *of* $m(X_n)$ *is universally Cesáro consistent.*

If, in addition, (A3) and

(N3) $\sum_{i=k_n+1}^{n-1} v_{ni} \to 0$ *as* $O(1/k_n^{1/2})$,

(N4) $\sup_n k_n \max_i v_{ni} < \infty$,

hold, then the rate of convergence of the time-averaged risks, $a_n^{(p)}$, *is*

$$O\left(\frac{1}{k_n^{1/2}} + \frac{k_n}{n} + \frac{1}{n}\sum_{i=k_n}^{n} [\mathcal{N}^{-1}(\lfloor i/k_n \rfloor, A)]^{\alpha}\right). \tag{5}$$

In particular, for $\mathbb{R}^r$, *the rate of convergence is*

$$O\left(1/k_n^{1/2} + (k_n/n)^{\alpha/r}\right).$$

Furthermore, there exists a k_n*-rule that achieves a rate of* $O(n^{-\frac{\alpha}{2\alpha+r}})$.

Typically the choice for v_{ni} are the uniform weights,

$$v_{ni} = \begin{cases} \frac{1}{k_n}, & i \le k_n \\ 0, & i > k_n \end{cases}$$

which obviously satisfy the hypothesis in our theorem as long as (N1) holds.

4.2 Kernel Regression Estimation

In this section we consider regression estimation using kernel estimates. For an arbitrary random process $\Omega_n = X_1, X_2, \ldots, X_n$, define the kernel weights

$$W_{ni}(\Omega_n) = \frac{\phi(\frac{\rho(X_i,X_n)}{\epsilon_n})}{\sum_{j=1}^{n-1} \phi(\frac{\rho(X_j,X_n)}{\epsilon_n})},$$

where $\{\epsilon_n\}$ is some sequence of positive numbers and $\phi(\cdot)$ is some arbitrary nonnegative measureable function, $\phi : \mathbb{R}_+ \to \mathbb{R}_+$, satisfying certain properties. Note that in the expression for the weights and in the sequel we treat 0/0 as 0. The kernel regression estimate is defined as

$$m_n(X_n) = \sum_{i=1}^{n-1} W_{ni}(\Omega_n) Y_i = \sum_{i=1}^{n-1} \frac{\phi(\frac{\rho(X_i,X_n)}{\epsilon_n}) Y_i}{\sum_{j=1}^{n-1} \phi(\frac{\rho(X_j,X_n)}{\epsilon_n})}.$$

which is motivated by the Parzen density function estimate. If the $\{X_i\}$ are i.i.d., it has been shown that under certain conditions $m_n(X_n) \to m(X_n)$ for either weak or strong consistency. Devroye and Wagner (1980) proved weak universal consistency of the kernel estimate. Devroye (1981) proved strong universal consistency under different conditions. Rates of convergence under certain assumptions were established by Krzyżak (1986). In Theorem 4 we show that similar conditions yield convergence for an arbitrary random process $\{X_i\}$, i.e., universal Cesáro consistency.

We first prove the next theorem which quantifies the clustering of points selected arbitrarily in a compact set. Define the number of arbitrarily chosen points in an ϵ_n-neighborhood of X_n as

$$S_n(\epsilon_n) = \sum_{i<n} I(X_i \in \mathcal{B}_{X_n,\epsilon_n}).$$

The following result is an analogue and extension of the well-known fact [see Devroye (1981)] that $S_n(\epsilon_n) \to \infty$ for almost every $\{X_i\}$ drawn i.i.d. in $\mathbb{R}^r$ as long as $n\epsilon_n^r \to \infty$. We extend the result to general metric spaces and allow $\{X_i\}$ to be any random process.

Theorem 3 *Let A be a compact subset of $(\mathcal{X}, \rho)$. For any random process $\{X_i\}_{i=1}^n \subset A$, any number $\gamma > 0$, and $\epsilon > 0$, we have a.s.*

$$\sum_{i=2}^{n} \left(\frac{I(S_i(\epsilon) \neq 0)}{S_i(\epsilon)} \right)^{\gamma} \leq \mathcal{N}(\epsilon/2, A) + \sum_{i=\mathcal{N}(\epsilon/2,A)}^{n-1} \left[\frac{2\mathcal{N}(\epsilon/2, A)}{i} \right]^{\gamma}.$$

Hence, if $\epsilon_n \to 0$ and is nonincreasing and $\mathcal{N}(\epsilon_n, A)/n \to 0$, we have

$$\frac{1}{n} \sum_{i=2}^{n} \left(\frac{I(S_i(\epsilon_n) \neq 0)}{S_i(\epsilon_n)} \right)^{\gamma} = \begin{cases} O\left(\left[\frac{\mathcal{N}(\epsilon_n, A)}{n} \right]^{\gamma} \right) \to 0, & 0 < \gamma < 1 \\ O\left(\frac{\mathcal{N}(\epsilon_n, A)}{n} \log \frac{n}{\mathcal{N}(\epsilon_n, A)} \right) \to 0, & \gamma = 1 \end{cases}$$

For example, with $A = [0, 1]^r$, then $\frac{1}{n} \sum_{i=2}^{n} \frac{1}{S_i(\epsilon_i)} \leq \frac{1}{n\epsilon_n^r} + \frac{2\log(n\epsilon_n^r)}{n\epsilon_n^r}$.

Theorem 4 *Let $\Omega_n = X_1, X_2, \ldots, X_n$ be an arbitrary random process taking values in a compact subset A of $(\mathcal{X}, \rho)$ with $(X_1, Y_1), \ldots, (X_n, Y_n)$ satisfying (A0). If the following conditions hold:*

(K1) $\epsilon_n \to 0$ and is nonincreasing,

(K2) $\mathcal{N}(\epsilon_n, A)/n \to 0$,

(K3) there exists constant $c_1, c_2, c > 0$ such that

$$c_1 I(u \leq c) \leq \phi(u) \leq c_2 I(u \leq c),$$

then the kernel estimate $m_n(X_n)$ of $m(X_n)$ is universally Cesáro consistent. If, in addition, (A3) holds, then the rate of convergence is

$$O\left(\frac{1}{n} \sum_{i=1}^{n} \epsilon_i^{\alpha} + (\mathcal{N}(\epsilon_n, A)/n)^{1/2} \right).$$

In particular, for $\mathbb{R}^r$, the rate of convergence is

$$O\left(\epsilon_n^{\alpha} + 1/n\epsilon_n^{r/2} \right),$$

and the rate $O(n^{-\alpha/(r+2\alpha)})$ can be achieved using $\epsilon_n = n^{-1/(r+2\alpha)}$.

4.3 Partitioning Estimates

For each n, let $P_n = \{A_{n1}, A_{n2}, \ldots\}$ be a partition of $(\mathcal{X}, \rho)$. Define $A_n(x)$ as the member of P_n to which x belongs. For our purposes we only consider sequences of partitions that are sequentially refined, that is, for each $A_{n+1,\,i}$ there exists j such that $A_{n+1,\,i} \subset A_{nj}$, or equivalently, that for each $x \in \mathcal{X}$, we have $A_{n+1}(x) \subset A_n(x)$.

As before, let $X_1, X_2, \ldots, X_n$ be an arbitrary random process taking values in a compact subset A of $(\mathcal{X}, \rho)$. The partitioning estimate (or histogram estimate) of $m(X_n)$ is

$$m_n(X_n) = \sum_{i=1}^{n-1} W_{ni}(\Omega_n) Y_i = \frac{\sum_{i=1}^{n-1} I(X_i \in A_n(X_n)) Y_i}{\sum_{i=1}^{n-1} I(X_i \in A_n(X_n))}.$$

In this expression we treat 0/0 as 0. The notion of using previous pairs of data to partition the space is the basis of the classification and regression trees methodology developed by Breiman, Friedman, Olshen, and Stone (1984). Gordon and Olshen (1980, 1984) proved the weak universal and almost sure consistency of recursive partitioning schemes. Devroye and Györfi (1983) addressed this estimate and proved that certain conditions imply weak universal consistency. Györfi (1991) proved the strong universal consistency. Rates of convergence using wavelets were obtained by Engel (1994).

We now show that similar conditions imply convergence for any sequence of X_i's, i.e., universal Cesáro consistency. Let $G_n(A) = \#\{i | A_{ni} \cap A \neq \emptyset\}$ where $\#S$ denotes the cardinality of a set S. Note that since P_n is a sequentially refined partition, $G_n(A)$ is nondecreasing. Let $T_n(B) = \sum_{i=1}^{n-1} I(X_i \in B)$ be the number of X_i's in any subset B of $(\mathcal{X}, \rho)$. Then $W_{ni}(\Omega_n) = \frac{I(X_i \in A_n(X_n))}{T_n(A_n(X_n))}$. The following lemma is analogous to Theorem 3.

Lemma 1 *Let A be a compact subset of $(\mathcal{X}, \rho)$ and let $P_n = \{A_{n1}, A_{n2}, \ldots\}$ be a sequence of sequentially refined partitions of $(\mathcal{X}, \rho)$. Then for any random process $X_1, X_2, \ldots, X_n \in A$ and $\gamma > 0$, we have a.s.*

$$\sum_{i=2}^{n} \left(\frac{I(T_i(A_n(x_i)) \neq 0)}{T_i(A_n(x_i))} \right)^{\gamma} \leq G_n(A) + \sum_{i=G_n(A)}^{n-1} \left(\frac{2G_n(A)}{i} \right)^{\gamma}.$$

Theorem 5 *Let $\Omega_n = X_1, X_2, \ldots, X_n$ be an arbitrary random process in a compact subset A of $(\mathcal{X}, \rho)$ with $(X_1, Y_1), \ldots, (X_n, Y_n)$ satisfying (A0). Let $P_n = \{A_{n1}, A_{n2}, \ldots\}$ be a sequence of sequentially refined partitions of $(\mathcal{X}, \rho)$. If the following conditions hold:*

(P1) $\max_{i: A_{ni} \cap A \neq \emptyset} \mathcal{N}^{-1}(1, A_{ni}) \to 0$,

(P2) $\frac{G_n(A)}{n} \to 0$,

then the partitioning estimate $m_n(X_n)$ of $m(X_n)$ is universally Cesáro consistent. If, in addition, (A3) holds, then the rate of convergence is

$$O\left(\left[\frac{G_n(A)}{n} \right]^{1/2} + \frac{1}{n} \sum_{i=1}^{n} \left[\max_{i: A_{ni} \cap A \neq \emptyset} \mathcal{N}^{-1}(1, A_{ni}) \right]^{\alpha} \right). \tag{6}$$

Remarks The conditions for consistency are very natural. (P1) means that the radii of the partition elements go to zero. (P2) limits the growth of the number of elements in the partition.

5 Stationary Processes

A form of Theorem 1 using expectation statements rather than almost sure statements can also be shown. With this form, the next theorem is a partial recovery and extension of the classical result of Stone (1977). Although the results are limited to continuous regression functions and to compact support of $\{X_i\}$ the results are extended to stationary processes in general metric spaces.

Theorem 6 *Let $\Omega_n = X_1, X_2 \ldots, X_n$ be a stationary process taking values in A, a compact subset of $(\mathcal{X}, \rho)$ with $(X_1, Y_1), \ldots, (X_n, Y_n)$ satisfying (A0). Let $\{\overline{W}_n\}$ be a sequence of weights. If (C1) and the following conditions hold:*

(S2) $E \sum_{i=1}^{n-1} |W_{ni}(\Omega_n)| I(d_{ni} > \epsilon) \to 0$, *for all* $\epsilon > 0$,

(S3) $E\left|\sum_{i=1}^{n-1} W_{ni}(\Omega_n) - 1\right| \to 0$,

(S4) $E \max_i |W_{ni}(\Omega_n)| \to 0$,

then the estimate $m_n(X_n) = \sum_{i=1}^{n-1} W_{ni}(\Omega_n) Y_i$ *satisfies*

$$E[r_n^{(p)}(\Omega_n)] \to 0 \;\; \forall p \geq 1$$

for all conditional distributions that satisfy (A1) and (A2).

Universal weak consistency and rates of convergence results for stationary processes can also be obtained for the specific estimators considered in Section 4 using our results for arbitrary processes. Pointwise consistency will be obtained using the stationarity of the $\{X_i\}$ and Theorem 6. Rates of convergence can be established by obtaining bounds on $E[d_n(k_n)]$, $E\left[\frac{1}{S_n(\epsilon_n)}\right]$, and $E\left[\frac{1}{T_n(A_n(X_n))}\right]$ under a stationarity assumption on the X_i's. The final result we get for the specific estimator is the following.

Corollary 1 *Let $X_1, \ldots, X_n$ be a stationary process taking values in a compact subset A of $(\mathcal{X}, \rho)$.*

1. *If $\{v_n\}$ satisfies (N1) and (N2), then the NN estimate satisfies $E[r_n^{(p)}(\Omega_n)] \to 0$ for all $p \geq 1$, for all conditional distributions that satisfy (A1) and (A2). Furthermore, if (A3),(N3), and (N4) hold then the rate of convergence is as in (5).*
2. *If $\{\overline{W}_n\}$ satisfies (K1) and (K2), then the kernel estimate satisfies $E[r_n^{(p)}(\Omega_n)] \to 0$ for all $p \geq 1$, for all conditional distributions that satisfy (A1) and (A2). Furthermore, if (A3) holds then the rate of convergence is as in (4).*
3. *If $\{\overline{W}_n\}$ satisfies (P1) and (P2), then the partitioning estimate satisfies $E[r_n^{(p)}(\Omega_n)] \to 0$ for all $p \geq 1$, for all conditional distributions that satisfy (A1) and (A2). Furthermore, if (A3) holds then the rate of convergence is as in (6).*

References

[1] Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J. (1984). *Classification and regression trees.* Wadsworth, Belmont.

[2] Cover, T.M. (1968). Estimation by the nearest neighbor rule. *IEEE Trans. Information Theory* **14** 50-55.

[3] Cover, T.M. and Hart, P.E. (1967). Nearest neighbor pattern classification. *IEEE Trans. Information Theory* **13** 21-27.

[4] Cover, T.M. (1991). Universal portfolios. *Mathematical Finance* **1** 1-29.

[5] Devroye, L. (1981). On the almost everywhere convergence of nonparametric regression function estimates. *Ann. Stat.* **9** 1310-1319.

[6] Devroye, L. (1982). Necessary and sufficient conditions for the pointwise convergence of nearest neighbor regression function estimates. *Z. Wahrscheinlichkeitstheorie verw. Gebiete* **61** 467-481.

[7] Devroye, L.P. and Györfi, L. (1983). Distribution free exponential bound for the $\mathcal{L}_1$ error of partitioning estimates of a regression function. *Probability and Statistical Decision Theory,* Proc. 4th Pannonian Symp. on Math. Stat., Bad Tatzmannsdorf, Austria.

[8] Devroye, L.P. and Wagner, T.J. (1980). Distribution-free consistency results in nonparametric discrimination and regression function estimation. *Ann. Stat.* **8** 231-239.

[9] Engel, J. (1994). A simple wavelet approach to nonparametric regression from recursive partitioning schemes. *J. Multivariate Anal.* **49** 242-254.

[10] Feder, M., Merhav, N., and Gutman, M. (1992). Universal prediction of individual sequences. *IEEE Trans. Information Theory* **38** 1258-1270.

[11] Gordon, L. and Olshen, R.A. (1980). Consistent nonparametric regression from recursive partitioning schemes. *J. Multivariate Anal.* **10** 611-627.

[12] Gordon, L. and Olshen, R.A. (1984). Almost surely consistent nonparametric regression from recursive partitioning schemes. *J. Multivariate Anal.* **15** 147-163.

[13] Greblicki, W., Krzyżak, A., and Pawlak, M. (1984). Distribution-free pointwise consistency of kernel regression estimate. *Ann. Stat.* **12** 1570-1575.

[14] Györfi, L. (1981a). The rate of convergence of k_n-NN regression estimates and classification rules. *IEEE Trans. Information Theory* **27** 362-364.

[15] Györfi, L. (1981b). Recent results on nonparametric regression estimate and multiple classification. *Prob. Contr. Inform. Theory* **10** 43-52.

[16] Györfi, L. (1991). Universal consistencies of a regression estimate for unbounded regression functions. *Nonparametric functional estimation and related topics*, 329-338, G. Roussas (ed.), Kluwer, Netherlands.

[17] Györfi, L., Härdle, W., Sarda, P., and Vieu, P. (1989). *Nonparametric curve estimation from time series.* Springer-Verlag, Berlin.

[18] Kolmogorov, A.N. and Tihomirov, V.M. (1961). ϵ-entropy and ϵ-capacity of sets in functional spaces. *American Mathematical Society Translations, series 2* **17** 277-364.

[19] Krzyżak, A. (1986). The rates of convergence of kernel regression estimates and classification rules. *IEEE Trans. Information Theory* **32** 668-679.

[20] Krzyżak, A. and Pawlak, M. (1984). Distribution-free consistency of a kernel regression estimate and classification. *IEEE Trans. Information Theory* **30** 78-81.

[21] Kulkarni, S.R. and S.E. Posner, (1994). Rates of convergence of nearest neighbor estimation under arbitrary sampling. *IEEE Trans. Information Theory*, 1028-1039, July.

[22] Royden, H.L. (1988). *Real Analysis.* Macmillan, New York.

[23] Stone, C.J. (1977). Consistent nonparametric regression. *Ann. Stat.* **5** 595-645.

[24] Stone, C.J. (1980). Optimal rates of convergence for nonparametric estimators. *Ann. Stat.* **8** 1348-1360.

[25] Stone, C.J. (1982). Optimal global rates of convergence for nonparametric regression. *Ann. Stat.* **10** 1040-1053.

[26] Truong, Y.K. and Stone, C.J. (1992). Nonparametric function estimation involving time series. *Ann. Stat.* **20** 77-97.

Analysis of an Approximate Decorrelating Detector

Narayan B. Mandayam† and Sergio Verdú‡

†WINLAB, Department of ECE, Rutgers University, Piscataway, NJ 08855-0909
‡Department of Electrical Engineering, Princeton University, Princeton, NJ 08544

Abstract

In this paper an approximate decorrelating detector is analyzed on the basis of a first order approximation to the inverse crosscorrelation matrix of signature waveforms. The approximation is fairly accurate for systems with low offset crosscorrelations and is exact in the two-user synchronous case. We present an exact as well as approximate analysis of the bit-error-rate performance of this detector on a channel that is subject to flat fading, and also specifically for the case of random signature waveforms. The detector outperforms the conventional matched filter receiver in terms of BER. The approximate decorrelator (while not being near-far resistant like the decorrelating detector) is fairly robust to imperfections in power control. Power trade-off regions are identified which characterize the significant advantage that the approximate decorrelator provides over the matched filter receiver. The reduced complexity of the approximate decorrelator and performance gain over the conventional matched filter makes it a viable alternative for implementation in real CDMA systems.

1 Introduction

Conventionally, demodulation of DS-CDMA signals is achieved with a matched filter receiver. These receivers are optimum only when there is no multiple access interference in additive Gaussian noise channels, and not only do they suffer from the near-far problem, but they have inferior performance even with perfect power control. Several multiuser detection schemes have been proposed (for eg: [1, 2]), however these are more complex than the matched filter detector and require explicit knowledge or estimates of the interference parameters. To overcome these deficiencies, adaptive detectors have been proposed recently (see [3] and references within) that are once again more complex than the matched filter detector to varying degrees. Our approach here is to derive a simplified multiuser detector that is only slightly more complex than the matched filter detector while still retaining some of the performance advantages of multiuser detectors. We consider a linear multiuser detector, namely the decorrelating detector [1] and analyze an approximation for it which was proposed in [4]. The detector performance on flat fading channels is characterized by the gain it offers over the matched filter in terms of BER, number of supportable users, and robustness to imperfections in power control. We derive approximate expressions for the above gains and show via simulations and exact analysis that these are fairly accurate over a wide range of system parameters. The simplicity of the analysis allows easy performance comparison with that of the matched filter.

1.1 Description of Detector

The approximate decorrelator that it is the subject of this paper was originally proposed in [4] for both the synchronous and asynchronous cases. For simplicity in illustration, we will consider the case of synchronous reception of the users' bits. In this case, the output of the matched filter for the k^{th} user can be written as

$$y_k = A_k b_k + \sum_{j \neq k} A_k b_j \rho_{j,k} + n_k, \tag{1}$$

where A_k, and b_k, are the amplitude and the bit of the k^{th} user respectively, and $n_k = \sigma \int_0^T n(t)s_k(t)dt$, with $s_k(t)$ being the signature waveform of the k^{th} user. The crosscorrelations between the waveforms is defined as

$$\rho_{jk} = \int_0^T s_j(t)s_k(t)dt. \tag{2}$$

The matched filter outputs for the users in the system can be expressed in vector form as

$$\mathbf{y} = \mathbf{RAb} + \mathbf{n}, \tag{3}$$

where $\mathbf{y} = [y_1 \cdots y_K]^\top$, $\mathbf{b} = [b_1 \cdots b_K]^\top$, $\mathbf{A} = \text{diag}\{A_1 \cdots A_K\}$, and $\mathbf{n}$ is a zero-mean Gaussian random vector with covariance matrix equal to $\sigma^2\mathbf{R}$. The matrix $\mathbf{R}$ of crosscorrelations among the user pairs is such that its ij^{th} entry is ρ_{ij}. The exact decorrelating detector is realized by subjecting the vector of matched filter outputs to a linear transformation as (for the k^{th} user)

$$\hat{b}_k = \text{sgn}(\mathbf{R^{-1}y})_k, \tag{4}$$

The above detector can be alternately realized by matched filtering the incoming signal with

$$\sum_{j=1}^{K} \mathbf{R}^{+}_{\mathbf{kj}} s_j(t), \tag{5}$$

where $\mathbf{R}^{+}_{\mathbf{kj}}$ denotes $(\mathbf{R}^{-1})_{kj}$. If, as is usually the case, the crosscorrelations among all the signal pairs are very low, $\mathbf{R}$ is strongly diagonal and (5) can be approximated by

$$s_k(t) - \sum_{j \neq k} \rho_{jk} s_j(t), \tag{6}$$

on the basis of $(\mathbf{I} + \delta\mathbf{L})^{-1} = \mathbf{I} - \delta\mathbf{L} + o(\delta)$. In fact, for the synchronous two-user case, the above expression is exact. The advantage of this approximation is greater in the asynchronous case, as in that case the crosscorrelations are not known in advance. In the remainder of the paper, we will consider the desired user to be user 1 and analyze the approximate decorrelator for synchronous and asynchronous channels.

2 Approximations for Bit-Error-Rate

In this section, we will analyze the approximate decorrelator for an asynchronous channel which is also subject to slow frequency non-selective Rayleigh fading. Consider the desired user taken to be user 1, and received signal $r(t)$ given as

$$r(t) = \alpha_1 A_1 b_{0,1} s_1(t) + \sum_{k=2}^{K} A_k(\alpha_{-1,k} b_{-1,k} s_k^L(t) + \alpha_{0,k} b_{0,k} s_k^R(t)) + \eta(t), \tag{7}$$

where the $\{\alpha_{i,j}\}$ are the (multiplicative) attenuation factors due to Rayleigh fading, $\{b_{i,j}\}$ are the bits of the respective users, and $s_k^L(t)$, $s_k^R(t)$ are the *left* and *right* signature waveforms due to the overlapping bits of the interferers. The simplified decorrelating detector structure for user 1 is given by matching the received signal to $h(t)$ given as

$$h(t) = s_1(t) - \sum_{k=2}^{K} \rho_{1k} s_k^L(t) - \sum_{k=2}^{K} \rho_{k1} s_k^R(t), \tag{8}$$

where the crosscorrelations $\{\rho_{ij}\}$ between every pair of signature waveforms depend on the offset between the signals and is given by

$$\rho_{ij}(\tau) = \int_0^T s_i(t)s_j(t-\tau),\ i<j, \tag{9}$$

and

$$\rho_{ji}(\tau) = \int_0^T s_i(t)s_j(t+T-\tau),\ i<j, \tag{10}$$

where $\tau \in [0,T]$. Note that for the conventional matched filter receiver $h(t) = s_1(t)$. The decision statistic, $y = \int_0^T r(t)h(t)$, is now given by

$$y = \alpha_1 A_1 b_{0,1} S + I_L + I_R + \mathcal{N}, \tag{11}$$

where

$$S = \int_0^T s_1(t)h(t), \tag{12}$$

$$I_L = \sum_{k=2}^{K} A_k \alpha_{-1,k} b_{-1,k} \int_0^T s_k^L(t)h(t), \tag{13}$$

$$I_R = \sum_{k=2}^{K} A_k \alpha_{0,k} b_{0,k} \int_0^T s_k^R(t)h(t), \tag{14}$$

and

$$\mathcal{N} = \int_0^T \eta(t)h(t), \tag{15}$$

with $\mathcal{N}$ being a zero mean Gaussian random variable with variance denoted by $\sigma_{\mathcal{N}}^2$. If $b_{0,1} = -1$, then the probability of error conditioned on $b_{0,1} = -1$ is given as

$$P_{e,-1} = E[Q(\frac{\delta}{\sigma_{\mathcal{N}}})], \tag{16}$$

where $\delta = \alpha_1 A_1 S - I_L - I_R$, and for $b_{0,1} = +1$, the above expression becomes

$$P_{e,+1} = E[Q(\frac{\gamma}{\sigma_{\mathcal{N}}})], \tag{17}$$

where $\gamma = \alpha_1 A_1 S + I_L + I_R$. The expectation in the above two expressions is over the interference and fading statistics $\mathcal{I} = (\alpha_1, \alpha_{-1,2}, \alpha_{0,2}, \cdots, \alpha_{-1,K}, \alpha_{0,K}, b_{-1,2}, b_{0,2}, \cdots b_{-1,K}, b_{0,K})$, and let us say it is characterized by a probability density function $f_{\mathcal{I}}$. The average probability of bit error (for equiprobable hypothesis) is now given by

$$P_e = \frac{1}{2}(P_{e,+1} + P_{e,-1}). \tag{18}$$

If the fading in the channel is deterministic, then an exact expression for the above probability of error can be evaluated as follows. Each bit of the desired user is affected by $2K-2$ interfering bits due to the interferers, and the probability of error in (18) is given as

$$P_e = \frac{1}{4^{K-1}} \sum_{b_{-1,2}\in B} \cdots \sum_{b_{-1,K}\in B} \sum_{b_{0,2}\in B} \cdots \sum_{b_{0,K}\in B} Q(\frac{A_1\alpha_1 S}{\sigma_{\mathcal{N}}} + \sum_{k=2}^{K} \frac{A_k}{\sigma_{\mathcal{N}}}(b_{-1,k}\alpha_{-1,k}\rho_k^L + b_{0,k}\alpha_{0,k}\rho_k^R)) \tag{19}$$

where $B = \{-1, +1\}$, and ρ_k^L, ρ_k^R are the correlated outputs due to the left and right signature waveforms and are given by

$$\rho_k^L = \int_0^T s_k^L(t)h(t), \tag{20}$$

and

$$\rho_k^R = \int_0^T s_k^R(t)h(t). \tag{21}$$

In deriving (19), we have assumed that the bits of all the users in the system are mutually independent. Thus we see that the exact expression for the probability of error can be evaluated although its complexity is increasing exponentially in the number of users in the system. Further, when the fading in the channel is random, the above expression will have to be averaged with respect to the fading statistics as well. This may be tedious, but can be accomplished with some degree of difficulty.

Owing to the analytical intractability in exactly evaluating the performance, we now derive an approximation for the bit-error-rate of the approximate decorrelator in an asynchronous channel with slow frequency non-selective Rayleigh fading. Without loss of generality, let us assume that the fading parameters $\{\alpha\}$ are i.i.d. random variables with $E[\alpha] = m$, and $Var(\alpha) = \sigma^2$. The bits of the interferers are also assumed to be i.i.d., with $E[b] = 0$, and $Var(b) = 1$. We also assume that the bits are independent of the fading parameters. Therefore, $E[I_L] = E[I_R] = 0$. The variance of I_L is now given as

$$Var(I_L) = \sum_{k=2}^{K} A_k^2 (y_k^L)^2 (\sigma^2 + m^2), \tag{22}$$

where once again we have used the independence assumption amongst users and fading parameters. In the above equation,

$$y_k^L = \int_0^T s_k^L(t)h(t). \tag{23}$$

A similar expression results for

$$Var(I_R) = \sum_{k=2}^{K} A_k^2 (y_k^R)^2 (\sigma^2 + m^2), \tag{24}$$

with

$$y_k^R = \int_0^T s_k^R(t)h(t). \tag{25}$$

Furthermore, it follows that $E[\gamma] = E[\delta] = mA_1S$, and

$$Var(\gamma) = Var(\delta) = A_1^2 S^2 \sigma^2 + Var(I_R) + Var(I_R). \tag{26}$$

Using just the first and second moments of the fading and interference statistics, an approximation for P_e is given as

$$P_e \approx \{\frac{2}{3}Q(\frac{E[\gamma]}{\sigma_{\mathcal{N}}}) + \frac{1}{6}Q(\frac{E[\gamma] + \sqrt{3Var(\gamma)}}{\sigma_{\mathcal{N}}}) + \frac{1}{6}Q(\frac{E[\gamma] - \sqrt{3Var(\gamma)}}{\sigma_{\mathcal{N}}})\}. \tag{27}$$

The details of the derivation of the above equation are given in the Appendix. In Figure 1, we analyze the bit-error-rate performance of the approximate decorrelator for an asynchronous system with fading. The simulations, exact analysis, and the approximations are shown for the approximate decorrelator. It is seen that the approximations derived in (27) are fairly accurate over a wide range of interferer power levels. As a comparison, the matched filter detector is also shown and it is observed that the approximate decorrelator outperforms the matched filter detector. As an example of the case with fading, the performance results for an asynchronous system is shown in Figure 2. We once again observe that the approximate decorrelator outperforms the matched filter receiver. The approximations for the bit-error-rate once again are fairly accurate compared to the simulations. The above results show that the approximation derived in (27) is fairly accurate over a wide range of interference power levels, background noise power levels, and the number of users in the system. Further, we can easily incorporate asynchronous users as well as fading in the analysis. Thus we have an approximation that is relatively simple in that it uses only means and variances of the relevant random variables in the system. In the remainder of the paper, we will work with the above approximation and do a comparative analysis of the advantages that the approximate decorrelator offers over a conventional matched filter receiver.

3 Analysis for Random Signature Waveforms

We will consider a synchronous DS-CDMA system employing random signature (PN) sequences, and analyze the performance of the approximate decorrelating detector in terms of bit-error-rate, power control, SNR gain, and sensitivity to imperfections in power control. Specifically, we will compare the relative performance to that of a conventional matched filter receiver. We will parametrize the performance of the two detectors in terms of the number of users K, the length of the random signatures N, the powers of the users $P_k (= A_k^2)$, and the noise variance $\sigma_{\mathcal{N}}^2$. We will set the fading coefficients $\{\alpha\}$ to be unity although the random fading case can be easily incorporated in the analysis.

3.1 BER Analysis

We will use the approximation in (27) to derive the bit-error-rates for both the conventional and the approximate decorrelating detector. In order to accomplish this, we need to derive the mean and variance of γ, and the corresponding noise variance. For the conventional matched filter receiver, we use the approximation (cf. Appendix)

$$P_e^{mf} \approx \{\frac{2}{3}Q(\frac{E[\gamma_{mf}]}{\sigma_{\mathcal{N}}^{mf}}) + \frac{1}{6}Q(\frac{E[\gamma_{mf}] + \sqrt{3Var(\gamma_{mf})}}{\sigma_{\mathcal{N}}^{mf}}) + \frac{1}{6}Q(\frac{E[\gamma_{mf}] - \sqrt{3Var(\gamma_{mf})}}{\sigma_{\mathcal{N}}^{mf}})\}, \quad (28)$$

where

$$E[\gamma_{mf}] = \sqrt{P_1}, \quad (29)$$

and

$$Var(\gamma_{mf}) = \frac{1}{N}\sum_{k=2}^{K} P_k, \quad (30)$$

and

$$\sigma_{\mathcal{N}}^{mf} = \sigma_{\mathcal{N}}. \quad (31)$$

The above relations are derived by setting $h(t) = s_1(t)$, and averaging over all the random signature sequences of length N, and comprising of $+1$ or -1 with equal probability. It is seen that γ_{mf} is given by

$$\gamma_{mf} = A_1 + \sum_{k=2}^{K} A_k b_k \rho_{1k}, \quad (32)$$

where ρ_{1k} is the crosscorrelation between user 1 and user k. Since $E[b_k] = 0$, it follows that $E[\gamma_{mf}] = A_1 = \sqrt{P_1}$. We will now derive the variance expression in (30) as follows.

$$Var(\gamma_{mf}) = Var(\sum_{k=2}^{K} A_k b_k \rho_{1k}) = E[(\sum_{k=2}^{K} A_k b_k \rho_{1k})^2],$$

which can be written as

$$Var(\gamma_{mf}) = E[\sum_{k=2}^{K} P_k \rho_{1k}^2 + \sum_{\substack{j,k \\ j\neq k}} A_k A_j b_k b_j \rho_{1k}\rho_{1j}] = \sum_{k=2}^{K} P_k E[\rho_{1k}^2],$$

where we have used the independence assumption among the transmitted bits. We will now derive an expression for $E[\rho_{1k}^2]$ from first principles as follows. Let $d_{ki} \in \{+1, -1\}$ denote the polarity of the i^{th} chip of the k^{th} user's signature sequence. Then by definition,

$$\rho_{jk} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{d_{ji} = d_{ki}\} - \mathbf{1}\{d_{ji} \neq d_{ki}\} = -1 + \frac{2}{N}\sum_{i=1}^{N} \mathbf{1}\{d_{ji} = d_{ki}\},$$

where the indicator function is defined as

$$\mathbf{1}\{a = b\} = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases}$$

Using the above relation, and by conditioning on the signature waveforms of the interferers and taking expectations with respect to the desired user, it can be shown that

$$E_1[\rho_{1k}\rho_{1l}] = \frac{\rho_{kl}}{N},$$

from which it follows that $E[\rho_{1k}^2] = 1/N$.

For the case of the approximate decorrelator, a similar approximation for the bit-error-rate results in

$$P_e^{ad} \approx \{\frac{2}{3}Q(\frac{E[\gamma_{ad}]}{\sigma_N^{ad}}) + \frac{1}{6}Q(\frac{E[\gamma_{ad}] + \sqrt{3Var(\gamma_{ad})}}{\sigma_N^{ad}}) + \frac{1}{6}Q(\frac{E[\gamma_{ad}] - \sqrt{3Var(\gamma_{ad})}}{\sigma_N^{ad}})\}, \quad (33)$$

where analogous to the matched filter case, the following expressions can be derived for the approximate decorrelating detector by setting $h(t) = s_1(t) - \sum_{k=2}^{K} \rho_{1k} s_k(t)$, resulting in

$$E[\gamma_{ad}] = \sqrt{P_1}(1 - \frac{K-1}{N}), \quad (34)$$

and

$$Var(\gamma_{ad}) = (\frac{(K-2)}{N^2} + \frac{(K-2)(K-3)}{N^3})(\sum_{k=2}^{K} P_k), \quad (35)$$

and

$$\sigma_N^{ad} = \sigma_N \sqrt{(1 + \frac{(K-1)(K-2)}{N^2} - \frac{(K-1)}{N})} \quad (36)$$

Thus, we have analytical expressions for the bit-error-rates of the matched filter (P_e^{mf}), and the approximate decorrelating detector (P_e^{ad}) as functions of the powers P_k, background noise variance σ_N^2, number of users K, and the length of the signature sequences N. The gain in BER offered by the approximate decorrelating detector over the matched filter detector is illustrated in Figure 3.

3.2 Power Trade-Off Regions for fixed BER

In this section, we characterize the robustness of the approximate decorrelator with respect to imperfections in power control. Since we have an analytical expression for the bit-error-rate of the detector, it can be inverted to yield the user powers (or alternately the signal-to-background noise) for a fixed level of bit-error-rate. Specifically, we characterize the power trade-off regions for both the matched filter and the approximate decorrelator for a fixed desired bit-error-rate. In Figure 4, we consider a system with $K = 30$ users, and concentrate on the effect of the variation of the power of user 2 on the other users. We assume that all users except user 2 have the same power. The power trade-off curves are plotted in terms of the $SNR's$ required for the user 1 (and users $k = 3, \cdots, 30$) and user 2 so that the users in the system achieve a bit-error-rate lesser than 10^{-3}. The exact decorrelator as well as the line of perfect power control are also shown for reference. The allowable operating region for the exact decorrelating detector is, as expected, insensitive to the imperfections in power control. It is seen that the approximate decorrelating detector is tolerant to a wider range of imperfections in power control as opposed to the matched filter detector which is intolerant to even slight imbalances in the respective powers of the users in the system.

3.3 Perfect Power Control Analysis

In this section, we will consider the case of perfect power control, and analyze the advantages the approximate decorrelator yields in terms of relative performance to the conventional matched filter receiver. Specifically, we consider the case of fixed bit-error-rate for both detectors and analyze the trade-off between SNR and the number of users that can be supported. First, we compare the gain offered by the approximate decorrelator in terms of number of users for a fixed SNR level over that of a matched filter receiver. In Figure 5, the number of users that can be supported in the system with a bit-error-rate of 10^{-3}, and a fixed level of SNR is shown for both the matched filter receiver and the approximate decorrelator. It is seen that the approximate decorrelator supports more than twice the number of users that a matched filter receiver can support for the same SNR level. An alternate comparison is shown in Figure 6, where for a fixed number of users the required SNR is shown for both the receivers to achieve a bit-error-rate of 10^{-3}. It is seen that the approximate decorrelator offers a SNR gain of several dBs over that of the matched filter receiver.

4 Conclusions

In this paper we have analyzed an approximate decorrelating detector derived on the basis of a first order approximation to the inverse crosscorrelation matrix of signature waveforms. The approximation is fairly accurate for systems with low offset crosscorrelations and is exact in the two-user synchronous case. An approximate analysis that results in simple expressions for the bit-error-rate of this detector was presented. These results were shown to be fairly accurate compared to an exact analysis as well as simulations. Specifically, for the case of random signature sequences, the detector was shown to outperform the conventional matched filter receiver in terms of gain in BER (for a fixed number of users) as well as gain in number of users (for a fixed level of performance). The approximate decorrelator (while not being near-far resistant like the decorrelating detector) was also seen to be fairly robust to imperfections in power control. Power trade-off regions were identified to characterize the significant advantage that the approximate decorrelator provides over the matched filter receiver. The advantages are multi-fold in terms of increased capacity, lower power requirements, and less stringent power control. The reduced complexity of the approximate decorrelator and performance gain over the conventional matched filter makes it a viable alternative for implementation in cellular CDMA systems, in particular in those systems where the period of the signature waveform is larger than the bit-period.

Appendix

We will now derive the approximation in (27). Consider

$$P_{e,+1} = \int Q(\frac{\gamma}{\sigma_{\mathcal{N}}}) f_{\Gamma}(\gamma) d\gamma = E[Q(\frac{\gamma}{\sigma_{\mathcal{N}}})], \tag{37}$$

where f_{Γ} is the probability density function of the interference and fading statistics. In other words, the random variable γ is distributed with the underlying density function being f_{Γ}, and let us denote its mean and variance, $E[\gamma]$ and $Var(\gamma)$, respectively. We will obtain an approximation for $P_{e,+1}$ in terms of just the mean and variance of γ as follows. Consider θ a random variable with mean μ, and variance σ^2, then assuming existence of derivatives, we can rewrite a function $P(\theta)$ using a Taylor series as follows

$$P(\theta) = P(\mu) + (\theta - \mu)P'(\mu) + \frac{1}{2}(\theta - \mu)^2 P''(\mu) + \cdots \tag{38}$$

By truncating the series to just terms of second order, and taking expectations one gets

$$E[P(\theta)] \approx P(\mu) + \frac{1}{2}P''(\mu)\sigma^2.$$

If instead of using a Taylor series, one uses an expansion in central differences (Stirling formula) [5], then we arrive at

$$E[P(\theta)] \approx P(\mu) + \frac{1}{2}\frac{P(\mu+h) - 2P(\mu) + P(\mu-h)}{h^2}\sigma^2$$

Actually, h should depend strongly on σ, the standard deviation of θ, since that defines where most of the variation of θ occurs. It was shown in [5], using the Gauss-Hermite quadrature [6], that $h = \sqrt{3}\sigma$ makes the approximation *exact* for fifth degree polynomials and normally distributed θ. It was also shown that the above approximation is fairly robust to non-Gaussian distributions and deviations from the above assumptions. Using the above approximation on (37) results in

$$P_{e,+1} \approx \{\frac{2}{3}Q(\frac{E[\gamma]}{\sigma_{\mathcal{N}}}) + \frac{1}{6}Q(\frac{E[\gamma] + \sqrt{3Var(\gamma)}}{\sigma_{\mathcal{N}}}) + \frac{1}{6}Q(\frac{E[\gamma] - \sqrt{3Var(\gamma)}}{\sigma_{\mathcal{N}}})\}. \tag{39}$$

Since $E[\gamma] = E[\delta]$, and, $Var(\gamma) = Var(\delta)$, the result in (27) follows from (18).

References

[1] R. Lupas and S. Verdu, "Linear multiuser detectors for synchronous code-division multiple-access channels," *IEEE Trans. Info. Theory*, vol. IT-34, pp. 123–136, Jan. 1989.

[2] M. K. Varanasi and B. Aazhang, "Multistage detection for asynchronous code-division multiple-access communications," *IEEE Trans. Commun.*, vol. COM-38, Apr. 1990.

[3] S. Verdu, "Adaptive Multiuser Detection", *Proceedings of IEEE International Symposium on Spread Spectrum Theory and Applications*, Oulu, Finland, July 1994.

[4] S. Verdu, "Multiuser Detection", *Advances in Statistical Signal Processing*, Vol. 2, pp. 369-409, JAI Press Inc.

[5] J.M. Holtzman, "On Using Perturbation Analysis to do Sensitivity Analysis: Derivatives vs. Differences" IEEE Trans. on Automatic Control, Vol. 37, No. 2, Feb. 1992, pp. 243-247.

[6] Z. Kopal, "Numerical Analysis", 2nd ed. New York: Wiley, 1964.

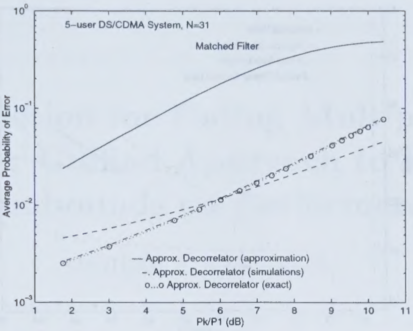


Figure 1: Performance of Approximate Decorrelating Detector

SNR (of the desired user) = 10 dB, $P_k = P_j$ for $j, k = 2, 3, 4, 5$

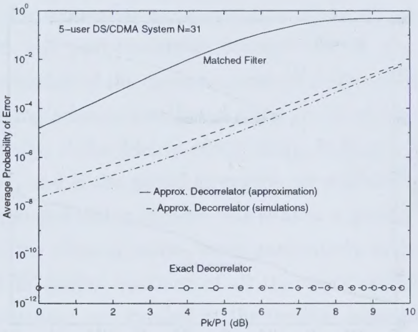


Figure 2: Performance of Approximate Decorrelating Detector (Fading)

SNR (of the desired user) = 20 dB, $P_k = P_j$ for $j, k = 2, 3, 4, 5$

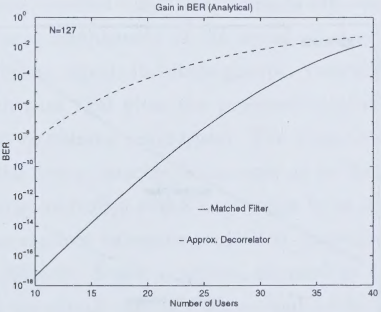


Figure 3: BER Gain

$P_k = P_1 = P$, and $P/\sigma_{\mathcal{N}}^2 = 20dB$

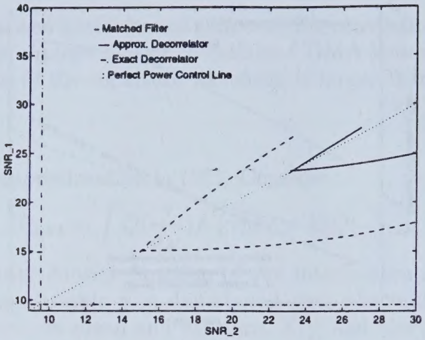


Figure 4: Power Trade-off Regions

$N = 127$, $K = 30$, and BER ≤ 0.001

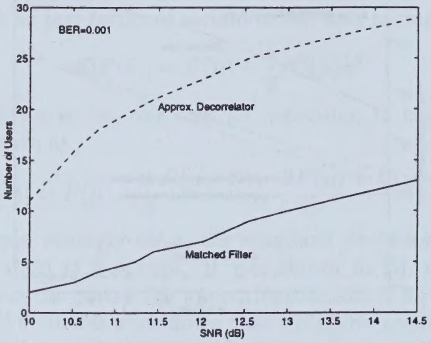


Figure 5: Gain in Number of Users (Perfect Power Control)

$N = 127$, BER $= 0.001$

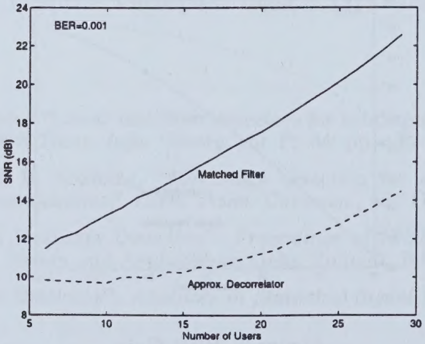


Figure 6: Gain in SNR (Perfect Power Control)

$N = 127$, BER $= 0.001$

Multiuser Detection for Fading Multipath Multiuser Channels: A Unified Approach to Design and Succinct Eigenbounds on Performance: Part II

MAHESH K. VARANASI†

Summary: The first part of this paper dealt with multiuser detection over fading multipath multiuser channels through a study of post-combining sequential group detection [1]. This part considers *pre-combining sequential group detection.* The pre-combining sequential group detector consists of a multiuser maximal ratio combiner as a front-end and is followed by a scheme that has the general structure of a sequential group detector as described in [2]. There are several differences between the pre- and post-combining schemes. In the pre-combining scheme, every group detector uses the knowledge of the channel states of all the channels, whereas, each group detector in the post-combining scheme uses the channel states of only the users in the current and past groups. Consequently, if the fidelity of the channel state estimates varies significantly enough with signal strength, and if the signal strengths are sufficiently disparate for this to be indeed the case, then the pre-combining scheme will exhibit a greater sensitivity to the imperfection in the knowledge of the channel states, most particularly to those of the weak users. If, on the other hand, the FSRF fading channels of all the users exhibit sufficiently slow fading, so that they can be well estimated and tracked at the receiver independently of signal strength, then the pre-combining scheme of this paper yields marked improvements in performance over the post-combining scheme. Hence, robust performance motivates post-combining and high performance motivates pre-combining. Another distinction between the two schemes is the way in which the pre-combining sequential group detector tunes out the signals from users in the future groups. It projects the residual received signal that remains after subtracting interference from past groups onto the orthogonal complement of the signal subspace spanned by the randomly time-varying aggregate interfering signals in future groups. It is this aggressive tracking of the random interfering signal subspace that gives the pre-combining scheme its significant performance advantage over its post-combining counterpart. The Time Complexity per Symbol (TCS) of the pre-combining sequential group detector is the same as for its post-combining counterpart and it can therefore be similarly controlled over a wide range by an appropriate choice of ordered group partitions. Finally, the random subspace projection operations make the analysis of the pre-combining scheme more difficult. Exact analytical expressions for asymptotic efficiency are not tractable. Nevertheless, we specify succinct upper and lower eigenbounds on the asymptotic efficiency of the pre-combining group detector that make the comparison between the pre- and post-combining group detectors transparent. Consequently, we also obtain upper and lower

bounds on the asymptotic efficiency performance of pre-combining sequential group detection. These are then compared with the corresponding bounds for post-combining sequential group detection.

1 System Model

The system model is the same as described in the first part of this paper. For the sake of self-containment however, we describe it once more. The received signal is modeled as a superposition of K digitally modulated waveforms that arrive at the receiver simultaneously after having passed through *slowly fading* frequency-selective Rayleigh fading channels. The coherence time of such channels is much greater than the data symbol duration so that accurate measurements of the channel characteristics are possible. Furthermore, the bandwidth assigned to each of the signature waveforms W is assumed to be much larger than the coherence bandwidth B_k of the k^{th} channel so that each channel can be represented as a tapped delay line with tap spacing $1/W$, and tap weight functions $\{c_{k,i}(t)\}_{i=0}^{L_k-1}$ [3]. These fading parameters represent the channel state information and are modeled as zero-mean complex-valued stationary Gaussian random processes. The number of resolvable paths L_k for the k^{th} signal is $\lceil W/B_k \rceil$, where $1/B_k$ is the k^{th} channel's multipath spread. The k^{th} user's signature waveform (possibly complex-valued) is time-limited to a symbol duration T, and is denoted as $u_k(t)$. The complex baseband representation of the signal that arrives at the receiver due to the isolated transmission of the modulated signature waveform of the k^{th} user is given as

$$h_k(t) = \sqrt{w_k} \sum_{i=0}^{L_k-1} c_{k,i}(t) u_k\left(t - \frac{i}{W}\right) = \sqrt{w_k}\mathbf{u}_k^T(t)\mathbf{c}_k(t), \tag{1}$$

where the k^{th} fading channel vector is defined as $\mathbf{c}_k(t) = [c_{k,0}(t), \cdots, c_{k,L_k-1}(t)]^T$, and the k^{th} user's signature waveform and its time-translates are defined by the vector

$$\mathbf{u}_k(t) = [u_k(t) \; u_k(t-1/W) \; \cdots \; u_k(t-(L_k-1)/W)]^T \; .$$

When a sequence of information symbols is transmitted, there is intersymbol interference (ISI). In wideband single-user communications where the symbol duration is much larger than the multipath spread, so that $L_k/W << T$, the ISI due to channel dispersion is usually ignored (cf. [3], [4]) in order to facilitate symbol-by-symbol decisions. In wideband multiuser communications however, there is ISI as well as inter-user interference (IUI), and even though ISI and IUI occur over a small fraction of a single symbol interval, the ever-present possibility of near-far conditions may accentuate their harmful effect, if ignored. Therefore, we choose to eliminate rather than ignore these interferences by masking out the received signal over time-intervals where both ISI and IUI are present [5]. The small loss in optimality is thereby traded for the simplicity of making

---

† The author is with the department of Electrical and Computer Engineering at the University of Colorado, Boulder, CO 80309. This work was supported by NSF grant NCR-9406069.

symbol-by-symbol decisions. The complex baseband representation of the received signal over the thusly restricted zeroth time-interval is given by

$$r(t) = \sum_{k=1}^{K} b_k h_k(t) + n(t) = \mathbf{h}^T(t)\mathbf{b} + n(t)\ , \qquad (L_M - 1)/W \leq t \leq T \tag{2}$$

where $\mathbf{b} = [b_1, \cdots, b_K]^T$ with b_k denoting the k^{th} user's information symbol and the signal vector $\mathbf{h}(t) = [h_1(t), \cdots, h_K(t)]^T$ and $L_M \triangleq \max L_k$. The circularly symmetric, complex, additive, and white, Gaussian noise process $n(t)$ has a power spectral density of height N_0. The symbol b_k belongs to F_k, an M-ary complex-valued symbol constellation of the k^{th} user.

A succinct description of the problem is facilitated by using the following block matrix notation. The L-length column vector (with $L \triangleq \sum_{k=1}^{K} L_k$) of signals obtained by stacking the vectors $\{\mathbf{u}_k(t)\}_{k=1}^{K}$ is denoted as $\mathbf{u}(t)$. The $L \times L$ signature signal correlation matrix is defined as

$$\mathcal{R} = \int_{(L_M-1)/W}^{T} \mathbf{u}^*(t)\mathbf{u}^T(t)dt = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} & \cdots & \mathbf{R}_{1K} \\ \mathbf{R}_{21} & \mathbf{R}_{22} & \cdots & \mathbf{R}_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{K1} & \mathbf{R}_{K2} & \cdots & \mathbf{R}_{KK} \end{bmatrix}, \tag{3}$$

with the superscripts $*$ and T denoting complex conjugation and matrix transposition, respectively. The matrix $\mathbf{R}_{kl}$ is the correlation matrix between the L_k- and the L_l-dimensional signal vectors $\mathbf{u}_k(t)$ and $\mathbf{u}_l(t)$, respectively, defined as $\mathbf{R}_{kl} = \int_{(L_M-1)/W}^{T} \mathbf{u}_k^*(t)\mathbf{u}_l^T(t)dt$. The signature signals are normalized to have unit energy and all their time-translates are assumed to be linearly independent so that $\mathcal{R}$ is positive definite. The diagonal matrix $\mathbf{W} = diag\{w_1, \cdots, w_K\}$ is such that w_k is the k^{th} user's total average received signal power, so that the expected value of $\int_{(L_M-1)/W}^{T} |\mathbf{c}_k^T(t)\mathbf{u}_k(t)|^2 dt = 1$. Let the covariance matrix of $\mathbf{c}_k$ be denoted as $\mathbf{\Sigma}_{kk}$. It is also convenient to define the $L \times K$ multipath multiuser channel state information matrix $\mathcal{C}$ as

$$\mathcal{C} = \begin{bmatrix} \mathbf{c}_1(t) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{c}_2(t) & \cdots & \mathbf{0} \\ & & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{c}_K(t) \end{bmatrix}, \tag{4}$$

where it is assumed that this matrix is essentially constant over a single symbol duration.

Let $\mathbf{q}$ denote the L-dimensional sufficient statistic vector that is obtained by stacking the L_k-dimensional vectors $\{\mathbf{q}_k\}_{k=1}^{K}$, where $\mathbf{q}_k$ is obtained as in [5] and defined as

$$\mathbf{q}_k = \int_{(L_M-1)/W}^{T} r(t)\mathbf{u}_k^*(t)\, dt\, . \tag{5}$$

The optimum pre-combining operation can be defined as one that reduces the L-dimensional sufficient statistic $\mathbf{q}$ to a K-dimensional minimal sufficient statistic $\mathbf{y}$ which results from matched filtering operations matched to the randomly time-varying aggregate signals $\{h_k(t) = \mathbf{c}_k^T\mathbf{u}_k(t)\}_{k=1}^{K}$. Equivalently, $\mathbf{y}$, the output of a bank of K randomly time-varying matched filters, can be generated from $\mathbf{q}$ from a bank of K single-user maximal ratio combiners to form $y_k = \mathbf{c}_k^T\mathbf{q}_k$. The following section describes pre-combining sequential group detection.

2 Pre-Combining Sequential Group Detection

The specification of the pre-combining sequential group detector and the results on performance analysis are presented here in a format that closely follows that of the post-combining sequential group detector in the first part of this paper. This is intended to bring out the similarities between the two schemes while making specific note of the important distinctions as and when they arise. It is not the intention of the author to lull the reader into an excessive sense of *déjà vu.* Be advised that the devil is in the detail.

Notation: Consider an ordered partition of the K users according to $\Omega = \{1, 2, \cdots, K\} = \cup_{j=1}^{J} G_j$. Without loss of generality, we let $G_j = \{K_{j-1}+1, \cdots, K_j\}$ with $K_0 = 0$ and $K_J = K$. Any arbitrary ordered partition of the K users can be expressed this way by an appropriate renumbering of the users so that users intended for group G_j would now be indexed from $K_{j-1}+1$ to K_j for $1 \leq j \leq J$.

Consider the K-dimensional vector of minimal sufficient statistics $\mathbf{y}$. We define several useful partitions of $\mathbf{y}$ that are induced by the ordered partition of users defined above. Let $\mathbf{y}^l$ be the vector of symbols of users in group G_l. Let $\mathbf{y}^{(m,j)}$ be obtained by stacking the vectors $\mathbf{y}_l$ for l ranging consecutively from m to j. Partitions of any vector that has the dimension of $\mathbf{y}$ such as $\mathbf{b}$, or even the vector of random signals $\mathbf{h}(t)$, are defined similarly. For instance, $\mathbf{h}^l(t)$ is the vector of the aggregate random signals of users in group G_l.

The diagonal matrix of average received signal powers $\mathbf{W}$ admits the obvious partition in terms of the group signal power matrices $\{\mathbf{W}^j\}_{j=1}^{J}$ as $\mathbf{W} = \text{diag}\{\mathbf{W}^1, \cdots, \mathbf{W}^J\}$ and we define the submatrix $\mathbf{W}^{(l,k)} = \text{diag}\{\mathbf{W}^l, \cdots, \mathbf{W}^k\}$. Define the $K \times K$ matrix S as being equal to $\mathcal{C}^{*T}\mathcal{R}\mathcal{C}$ and let the block partitioned form that is consistent with the ordered group partition be given as

$$\mathsf{S} = \begin{bmatrix} \mathsf{S}^{11} & \mathsf{S}^{12} & \cdots & \mathsf{S}^{1J} \\ \mathsf{S}^{21} & \mathsf{S}^{22} & \cdots & \mathsf{S}^{2J} \\ \vdots & \vdots & \ddots & \vdots \\ \mathsf{S}^{J1} & \mathsf{S}^{J2} & \cdots & \mathsf{S}^{JJ} \end{bmatrix}, \tag{6}$$

where the $|G_i| \times |G_j|$ matrix S^{ij} is the random correlation matrix between the aggregate random signal vectors $\mathbf{h}^i(t)$ and $\mathbf{h}^j(t)$ of groups G_i and G_j. Furthermore, define the submatrix of S obtained by extracting block rows k to l and block columns p to q in the above representation as $\mathcal{S}^{(k,l)(p,q)}$ so that

$$\mathcal{S}^{(k,l)(p,q)} = \begin{bmatrix} \mathsf{S}^{kp} & \cdots & \mathsf{S}^{kq} \\ \vdots & \ddots & \vdots \\ \mathsf{S}^{lp} & \cdots & \mathsf{S}^{lq} \end{bmatrix}. \tag{7}$$

Finally, we note that partitions of matrices with the dimension of S are defined similarly. We are now ready to describe the pre-combining sequential group detector.

PRE-COMBINING SEQUENTIAL GROUP DETECTOR

The first step involves the pre-combining group detection of users in the first group. The second and the third steps describe the interference cancellation process followed by the group detection operation. These steps are performed sequentially starting with $j = 2$ and terminating at $j = J$.

1. *Group detection for the first group:* An optimum detection of symbols in the first group G_1 is obtained after projecting the received signal onto the orthogonal complement of the subspace spanned by the randomly time-varying aggregate signals of the users in future groups $G_2, \cdots, G_J$ [6]. This rule is denoted as ψ_{G_1} and is described by

$$\psi_{G_1}: \quad \hat{\mathbf{b}}^1 = \arg\max_{\mathbf{x}\in F^1} 2\mathrm{Re}\left(\mathbf{x}^{*T}\left[\mathbf{W}^1\right]^{1/2}\mathbf{V}^{11}\left[\mathbf{S}^{-1}\mathbf{y}\right]^1\right) - \mathbf{x}^{*T}\left[\mathbf{W}^1\right]^{1/2}\mathbf{V}^{11}\left[\mathbf{W}^1\right]^{1/2}\mathbf{x}, \tag{8}$$

   where $\mathbf{V}^{11} \triangleq [[\mathbf{S}^{-1}]^{11}]^{-1}$ and F^1 denotes the joint alphabet of the symbols $\mathbf{b}^1$.

2. *Interference Cancellation:* The multiuser interference contributed by users in the previously detected groups $G_1, \cdots, G_{j-1}$ to the matched filter output vectors of users in the current and future groups $G_j, \cdots, G_J$ is estimated and subtracted,

$$\mathbf{r}^{(j)} = \mathbf{y}^{(j,J)} - \mathbf{S}^{(j,J)(1,j-1)}\left[\mathbf{W}^{(1,j-1)}\right]^{1/2}\hat{\mathbf{b}}^{(1,j-1)}. \tag{9}$$

   With the assumption of perfect interference cancellation, there is no loss of sufficiency in restricting attention to $\mathbf{y}^{(j,J)}$.

3. *Sequential Group Detection for G_j:* The symbols of users in group G_j are detected optimally after tuning out users in future groups $G_{j+1}, \cdots, G_J$ by projecting the residual received signal, that remains after subtracting the interference from past groups, onto the orthogonal complement of the subspace spanned by the aggregate interfering signals in the future groups. Consequently, the j^{th} sequential group detector is defined as

$$\tilde{\psi}_{G_j}: \quad \hat{\mathbf{b}}^j = \arg\max_{\mathbf{x}\in F^j} 2\mathrm{Re}\left(\mathbf{x}^{*T}\left[\mathbf{W}^j\right]^{1/2}\mathbf{V}^{jj}\left[\mathbf{U}^{(j)}\mathbf{r}^{(j)}\right]^1\right) - \mathbf{x}^{*T}\left[\mathbf{W}^j\right]^{1/2}\mathbf{V}^{jj}\left[\mathbf{W}^j\right]^{1/2}\mathbf{x} \tag{10}$$

   where F^j is the joint alphabet of $\mathbf{b}^j$ and $\mathbf{U}^{(j)} = \left[\mathbf{S}^{(j,J)(j,J)}\right]^{-1}$. Furthermore, we define $\mathbf{V}^{jj}$ as the inverse of $\left[\mathbf{U}^{(j)}\right]^{(1,1)}$ which is the first diagonal block of dimension $|G_j| \times |G_j|$ of the block matrix representation of $\mathbf{U}^{(j)}$. Also, $\left[\mathbf{U}^{(j)}\mathbf{r}^{(j)}\right]^{(1)}$ represents the first subvector of dimension $|G_j|$ in the natural block vector representation of $\mathbf{U}^{(j)}\mathbf{r}^{(j)}$.

The complexity of the above sequential group detector is determined by its TCS which is given as $O(\sum_{j=1}^{J} M^{|G_j|})$ for M-ary signalling with $|G_j|$ being the j^{th} group size. The sequential group detector yields a combiner-decorrelator based decision feedback detector when the group sizes are all equal to one. The combiner-decorrelator was introduced in [6] and the decision feedback detector based on it can be regarded as another generalization to the multipath fading multiuser channel of the decorrelating decision feedback detector of the Gaussian multiuser channel (cf. [7] [2]). Moreover, the sequential group detector reduces to the optimum detector [5] for the trivial partition where all users are in one group of size K.

3 Conditional Asymptotic Efficiency

Definition: The *conditional asymptotic efficiency* of any user k employing the detector ϕ whose conditional bit error probability for that user is $P_k(\mathbf{w}, N_0, \phi/\mathcal{C})$, is formally defined as

$$\eta_k(\phi/\mathcal{C}) \triangleq \sup \left\{ 0 \leq r \leq 1; \quad \lim_{N_0 \to 0} \frac{P_k(\mathbf{w}, N_0, \phi/\mathcal{C})}{Q\left(\sqrt{\frac{2rw_k \mathbf{c}_k^{*T} \mathbf{R}_{kk} \mathbf{c}_k}{N_0}}\right)} < \infty \right\}, \tag{11}$$

with the denominator being the minimum conditional bit error probability achievable in a single user channel where the k^{th} user transmits in isolation and has an average received signal energy per symbol that is equal to rw_k.

Proposition 1 *Under the assumption of perfect multiuser interference cancellation, the sequential group detector $\tilde{\psi}_{G_j}$ has the highest worst-case conditional asymptotic efficiency for every user in G_j—with the worst-case being taken over the average signal powers of all users in future groups $G_{j+1}, \cdots, G_J$—that is achievable by any detector that has perfect side-information about symbols of users in previous groups $G_1, \cdots, G_{j-1}$ (this class of genie-aided detectors is denoted as Ψ_j), which is the minimax optimality result:*

$$\tilde{\psi}_{G_j} \in \arg \max_{\phi \in \Psi_j} \inf_{\mathbf{W}^{(j+1,J)}} \eta_k(\phi/\mathcal{C}), \tag{12}$$

where max *denotes element-wise maximum, i.e., the pre-combining sequential group detector maximizes the worst-case conditional asymptotic efficiency for every user in group G_j.*

Notice that the infimum of the conditional asymptotic efficiency in the above minimax optimality result is considered over the average signal powers of users in future groups. This must be compared to the corresponding result in [1] where the post-combining sequential group detector admits a similar proposition with the distinction that the infimum is taken over the signal powers as well as the amplitudes of all faded paths of those users. As a result, the pre-combining sequential group detector proposed in this paper optimizes a much finer measure of conditional performance and is therefore likely to far out-perform its post-combining counterpart.

From a detector design point of view, the conditional nature of the above performance measure is ideal since the optimality in this measure is assured for every realization of $\mathcal{C}$, provided of course that the ordered partition can be chosen such that the assumption of perfect cancellation is in fact valid. From an analysis view-point however, it must be noted that the expected asymptotic efficiency, defined as the expected value of the conditional asymptotic efficiency, may not be a faithful measure of the expected bit-error rate.

4 Asymptotic Efficiency

Let the average bit error probability of a detector ϕ for the k^{th} user in a FSRF-CDMA channel be denoted as $P_k(\mathbf{w}, N_0, \phi)$, where $\mathbf{w}$ is the vector of average received signal power of the K

users. Let $P_k^{SU}(w, N_0)$ be the average bit-error rate of the single-user RAKE receiver operating in the k^{th} single-user FSRF channel where the k^{th} user transmits in isolation and whose signal at the receiver has average signal power equal to w.

Definition: The *asymptotic efficiency* of a detector ϕ operating in the FSRF-CDMA channel is formally defined as

$$\eta_k(\phi) = \sup\left\{0 \le r \le 1;\ \lim_{N_0 \to 0} \frac{P_k(\mathbf{w}, N_0, \phi)}{P_{k,SU}(rw_k, N_0)} \le 1\right\}. \tag{13}$$

This measure captures the loss in bit-error rate (BER) performance of a detector in the high SNR region where the dominant degradation in performance is due to the presence of interfering users rather than additive noise. In particular, a detector with a non-zero value of asymptotic efficiency $\eta_k(\phi)$ for user k operating in high SNR conditions has a BER performance that rivals the BER of the single-user RAKE receiver designed for the k^{th} single user FSRF channel with average received SNR per bit that is equal to $w_k \eta_k(\phi)$.

We first state the result for upper and lower bounds on the asymptotic efficiency of users in the first group because the first group detector ψ_{G_1} is conceptually simpler than the subsequent detectors $\tilde{\psi}_{G_j}$. It does not involve the interference cancellation step. For convenience, let us consider the group variable $G = \{1, \cdots, |G|\}$. Let $\mathbf{F}_k(G)$ be the correlation matrix of the signal vector obtained by stacking $\mathbf{u}_k(t)$ and the interfering signal vector $\mathbf{z}(t)$ which in turn is defined as being the result of stacking signal vectors of users $|G|+1$ to K. Writing this correlation matrix as a 2×2 block partitioned matrix, we have

$$\mathbf{F}_k(G) \triangleq \int_{(L_M-1)/W}^{T} \begin{pmatrix} \mathbf{u}_k(t) \\ \mathbf{z}(t) \end{pmatrix}^* \begin{pmatrix} \mathbf{u}_k(t) \\ \mathbf{z}(t) \end{pmatrix}^T dt = \begin{bmatrix} \mathbf{X}_k(G) & \mathbf{Y}_k^{*T}(G) \\ \mathbf{Y}_k(G) & \mathbf{Z}_k(G) \end{bmatrix}, \tag{14}$$

with $\mathbf{z}(t) \triangleq [\mathbf{u}_{|G|+1}^T(t), \cdots, \mathbf{u}_K^T(t)]^T$ is an interfering signal vector of length $\sum_{l=|G|+1}^{K} L_l$. It is remarkable that the asymptotic efficiency of the pre-combining group detector admits bounds that are functions of the eigenvalues of a matrix that can be described in terms of the components in the above block partitioned matrix. These are the same matrices that were used in stating the asymptotic efficiency of the post-combining group detector.

Proposition 2 *The asymptotic efficiency of the first group detector ψ_{G_1} for the k^{th} user in G_1 admits the upper bound given as*

$$\eta_k(\psi_{G_1}) \le \left(\prod_{i=1}^{K-K_1} \lambda_i\left(\mathbf{I} - [\mathbf{Z}_k(G_1)]^{-1}\mathbf{Y}_k(G_1)[\mathbf{X}_k(G_1)]^{-1}\mathbf{Y}_k^{*T}(G_1)\right)\right)^{1/L_k}, \tag{15}$$

where $\{\lambda_i(\mathbf{A})\}$ denote the eigenvalues of the matrix $\mathbf{A}$ arranged in **decreasing** *order.*

The following proposition states the lower bound on the asymptotic efficiency which is greater than or equal to the asymptotic efficiency of the post-combining group detector described in [1].

Proposition 3 *The asymptotic efficiency of the first group detector* ψ_{G_1} *for the* k^{th} *user in* G_1 *admits lower bounds given as*

$$\eta_k(\psi_{G_1}) \geq \left(\prod_{i=1}^{K-K_1} \delta_i \left(\mathbf{I} - [\mathbf{Z}_k(G_1)]^{-1} \mathbf{Y}_k(G_1) [\mathbf{X}_k(G_1)]^{-1} \mathbf{Y}_k^{*T}(G_1) \right) \right)^{1/L_k} \geq \eta_k(\phi_{G_1}), \tag{16}$$

where $\{\delta_i(\mathbf{A})\}$ *denote the eigenvalues of the matrix* $\mathbf{A}$ *arranged in* **increasing** *order. The weaker of the two lower bounds is equal to the asymptotic efficiency of the post-combining group detector* ϕ_G *and was shown to admit the formula given by*

$$\eta_k(\phi_{G_1}) = \left\langle \left\{ \lambda_i \left(\mathbf{I} - [\mathbf{X}_k(G_1)]^{-1} \mathbf{Y}_k^{*T}(G_1) [\mathbf{Z}_k(G_1)]^{-1} \mathbf{Y}_k(G_1) \right) \right\}_{i=1}^{L_k} \right\rangle_{\mathcal{G}} . \tag{17}$$

The weaker lower bound of the above proposition was proved in [6]. The new lower bound can be shown to improve upon the post-combining group detector bound when the number of paths of the user in question is greater than the number of interferers. The two lower bounds coincide otherwise. The following corollary illustrates the significant improvement in performance that can be achieved by the pre-combining group detectors over their post-combining counterparts by stating the special case of Propositions 2 and 3 for the combiner-decorrelator $\psi_{\{1\}}$ in the two-user case.

Corollary 1 *The asymptotic efficiency of the combiner-decorrelator for the first user of a two-user channel is bounded from above and below as*

$$\left(\lambda_{min} \left(\mathbf{I} - \mathbf{R}_{22}^{-1} \mathbf{R}_{21} \mathbf{R}_{11}^{-1} \mathbf{R}_{12} \right) \right)^{1/L_1} \leq \eta_1 \left(\psi_{\{1\}} \right) \leq \left(\lambda_{max} \left(\mathbf{I} - \mathbf{R}_{22}^{-1} \mathbf{R}_{21} \mathbf{R}_{11}^{-1} \mathbf{R}_{12} \right) \right)^{1/L_1} , \tag{18}$$

whereas the asymptotic efficiency of the decorrelator-combiner for the first user of a two-user channel is given by

$$\eta_1 \left(\phi_{\{1\}} \right) = \left\langle \left\{ \lambda_i \left(\mathbf{I} - \mathbf{R}_{22}^{-1} \mathbf{R}_{21} \mathbf{R}_{11}^{-1} \mathbf{R}_{12} \right) \right\}_{i=1}^{L_1} \right\rangle_{\mathcal{G}} . \tag{19}$$

Furthermore, the eigenvalues of the matrix $\mathbf{I} - \mathbf{R}_{22}^{-1} \mathbf{R}_{21} \mathbf{R}_{11}^{-1} \mathbf{R}_{12}$ *lie in the interval* $[0, 1]$.

Next, we consider the performance analysis of the sequential group detector $\tilde{\psi}_{G_j}$ which is more complicated than that of the group detector ψ_{G_1} because the former involves interference cancellation. Define super-groups $S_m = \cup_{l=1}^{m} G_l$ for $1 \leq m \leq J$. Consider the super-group detector ψ_{S_m} which is the group detector for users in the super-group S_m. Let p denote the i^{th} user in group G_m, so that $p = i + K_{m-1}$. The upper bound on the asymptotic efficiency of this (non-sequential) super-group detector bounds from above the performance of the sequential group detector $\tilde{\psi}_{G_m}$ defined in (10) for each user $i \in G_m$. This is the result of the following proposition. Before we state it, we let the upper bound of Proposition 2 of the pre-combining group detector ψ_G for user $k \in G$ be denoted as $\eta_{k,UB}(\psi_G)$.

Proposition 4 *The asymptotic efficiency of the* m^{th} *sequential group detector* $\tilde{\psi}_{G_m}$ *for the* i^{th} *user of group* G_m *(i.e., for user* $p = i + K_{m-1}$*) is upper bounded by the upper bound on the* *asymptotic efficiency of the super-group detector* ψ_{S_m} *for that user, so that*

$$\eta_p(\tilde{\psi}_{G_m}) \leq \eta_{p,UB}(\psi_{S_m}) \triangleq \left(\prod_{i=1}^{K-K_m} \lambda_i \left(\mathbf{I} - [\mathbf{Z}_p(S_m)]^{-1} \mathbf{Y}_p(S_m) [\mathbf{X}_p(S_m)]^{-1} \mathbf{Y}_p^{*T}(S_m) \right) \right)^{1/L_p} , \quad (20)$$

where the matrices involved are defined as in (14) with k *replaced by* p *and the group variable* G *replaced by the super-group* S_m*.*

The next proposition specifies a lower bound on the asymptotic efficiency and thereby includes the deleterious effects of error propagation that can result from interference enhancement rather than interference cancellation due to errors in previous groups. It gives rise to a set of sufficient conditions where the sequential group detector is guaranteed to have an asymptotic efficiency that is at least as high as that of the lower bound on the super group detector asymptotic efficiency. Before we state it however, we let the tighter lower bound of Proposition 3 of the pre-combining group detector ψ_G for user $k \in G$ be denoted as $\eta_{k,LB}(\psi_G)$.

Proposition 5 *The asymptotic efficiency of the pre-combining sequential group detector* $\tilde{\psi}_{G_j}$ *for the* k^{th} *user of group* G_j *(or the* q^{th} *user where* $q = k + K_{j-1}$*) is lower bounded as*

$$\eta_q(\tilde{\psi}_{G_j}) \geq \min \left\{ \min_{1 \leq m \leq j-1} \ \min_{p \in G_m; s.t. L_p \leq L_q} \ \gamma_p \frac{w_p}{w_q} \eta_{p,LB}(\psi_{S_m}) \ , \ \eta_{q,LB}(\psi_{S_j}) \right\} , \quad (21)$$

where $\eta_{p,LB}(\psi_{S_m})$ *is the tighter lower bound on the super-group detector performance specified in Proposition 4, and where the function* γ_p *is defined as*

$$\gamma_p = \begin{cases} 0 & \text{if } L_p < L_q ; \\ \left\langle \left\{ \lambda_l (\mathbf{\Sigma}_{pp} \mathbf{\Sigma}_{qq}^{-1} \mathbf{R}_{pp} \mathbf{R}_{qq}^{-1}) \right\}_{l=1}^{L_q} \right\rangle_{\mathcal{G}} & \text{if } L_p = L_q \ . \end{cases} \quad (22)$$

Consequently, when the number of paths of all users are equal, a set of sufficient conditions for the sequential group detector $\tilde{\psi}_{G_j}$ *to have a performance that is at least as high as the super-group detector lower bound for user* $q \in G_j$ *are as follows:*

$$\eta_q(\tilde{\psi}_{G_j}) \geq \eta_{q,LB}(\psi_{S_j}) \quad \text{if} \quad \frac{w_p}{w_q} \geq \frac{\eta_{q,LB}(\psi_{S_j})}{\gamma_p \eta_{p,LB}(\psi_{S_m})} \qquad \forall 1 \leq p \leq K_{j-1} . \quad (23)$$

The sufficient conditions in the above proposition suggests ways of selecting group sizes and the ordering of the groups so as to eliminate/mitigate the effects of error propagation. Note that the above proposition doesn't lend itself to being easily compared with its counterpart for the post-combining group detector. However, it is strong enough to give rise to the following corollary in which the comparison between the pre- and post-combining sequential group detection schemes is more evident.

Corollary 2 *When the number of paths of all users are equal, a set of sufficient conditions for the sequential group detector* $\tilde{\psi}_{G_j}$ *to have an asymptotic efficiency that is at least as high as the post-combining super-group detector for user* $q \in G_j$ *are as follows:*

$$\eta_q(\tilde{\psi}_{G_j}) \geq \eta_q(\phi_{S_j}) \quad \text{if} \quad \frac{w_p}{w_q} \geq \frac{\eta_q(\phi_{S_j})}{\gamma_p \eta_{p,LB}(\psi_{S_m})} \qquad \forall 1 \leq p \leq K_{j-1} \quad (24)$$

The interpretation of the above result is as follows. Note from Proposition 3 of [1] that $\eta_q(\phi_{S_j})$ is an upper bound on the asymptotic efficiency of the post-combining sequential group detector $\tilde{\phi}_{G_j}$. Consider also the fact that the sufficient conditions of the above corollary are "better" than those stated in Proposition 4 of [1] that in turn ensure that the upper bound $\eta_q(\phi_{S_j})$ is met by the detector $\tilde{\phi}_{G_j}$ with equality. By "better", we mean that if the sufficient conditions in Proposition 4 of [1] are met, then so are the sufficient conditions given in the above result. The converse of this statement is not necessarily true. Taken together with the above corollary, we conclude that pre-combining sequential group detector $\tilde{\psi}_{G_j}$ has a performance that is at least as high as the post-combining sequential group detector $\tilde{\phi}_{G_j}$ for each user in G_j when the sufficient conditions of stated in Proposition 4 of [1] are satisfied. An even stronger result is that there exist average signal power distributions (those that satisfy the sufficient conditions of the above corollary but not of Proposition 4 of [1]) where the pre-combining sequential group detector $\tilde{\psi}_{G_j}$ has a performance that is at least as high as the upper bound on the post-combining sequential group detector that is not necessarily achieved by the latter. Numerical results are awaited.

References

[1] M. K. Varanasi, "Multiuser Detection in Frequency-Selective Rayleigh Fading Channels: A Unified Approach to Design and Succinct Eigenbounds on Performance: Part I," *this Proc. of the Annual Allerton Conf on Communication, Control, and Computing*, University of Illinois, Urbana-Champaign, Oct 2-4, 1995.

[2] M. K. Varanasi, "Group Detection for Synchronous Gaussian Code-Division Multiple-Access Channels," *IEEE Trans Inform. Th.*, **41**:4 (July 1995).

[3] J. G. Proakis, "*Digital Communications*," Third Ed., Mc-Graw Hill, NY, 1995.

[4] M. Schwartz, W. R. Bennett and S. Stein, *Communication Systems and Techniques*," Inter-University Electronics Series, Vol-4, McGraw-Hill Book Company, 1966.

[5] M. K. Varanasi, "Parallel Group Detection for Synchronous Synchronous CDMA Communication Over Frequency-Selective Rayleigh Fading Channels," *IEEE Trans Inform. Th.*, **43**:1 (January 1996).

[6] M. K. Varanasi, "Group Detection for Synchronous CDMA Communication Over Frequency-Selective Fading Channels," *Proc 31st Annual Allerton Conf on Communication, Control, and Computing*, Sep 29-Oct 1, 1993, pp. 849-858.

[7] A. Duel-Hallen, "Decorrelating Decision Feedback Multiuser Detector for Synchronous Code-Division Multiple-Access Channel," *IEEE Trans Commun* **COM-41**:2, 285-290, February 1993.

Two-stage Multiuser Detection for Noncoherent CDMA

Chris Hegarty
Center for Advanced Aviation System Development
MITRE Corporation

Branimir Vojcic
EE&CS Department
George Washington University

Abstract

In this paper, a two-stage multiuser detector for noncoherent CDMA is proposed and analyzed. The detector operates using tentative bit decisions of the bilinear decorrelating detector to estimate the multiple access interference corrupting the desired signal. The two-stage detector subsequently subtracts the reconstructed interference from the received signal, and then forms final bit estimates using a bank of conventional differential phase shift keying (DPSK) detectors. Numerical examples indicate that the two-stage detector may offer a performance advantage over the bilinear decorrelator with little added complexity.

Introduction

In recent years, there has been a great deal of research performed to investigate methods of alleviating the performance degradations suffered by code division multiple access (CDMA) systems due to multiple access interference (MAI). One of the seminal works in this area was the optimal multiuser detector for coherent, asynchronous CDMA developed by Verdu [1]. Although this detector provides optimal performance in terms of aggregate bit error rate, it has the drawback that its complexity grows exponentially with the number of users. Detailed performance analyses of linearly decorrelating detectors followed for both synchronous [2] and asynchronous [3] coherent CDMA which offer linear complexity with the number of users at the cost of a sacrifice in performance. Multistage detectors were then developed [4] which retain the linear complexity attribute of the decorrelating detectors, but perform closer to optimal.

Work in the area of multiuser detection has also extended to noncoherent channels. Bilinear decorrelating detectors have been proposed and analyzed for both the synchronous [5] and asynchronous differentially coherent channel [6]. The bilinear decorrelating detector closely parallels its coherent counterpart in both its formulation and performance analysis. A question which naturally arises at this point is whether a multistage architecture similar to that used for the coherent case would also provide some performance benefits over the noncoherent decorrelating detector. This is indeed the case as is shown in the sequel through the development and analysis of a two-stage noncoherent detector for synchronous CDMA.

System Model

The complex envelope, r(t), of the received signal of a synchronous K-user non-coherent CDMA system may be expressed as:

$$r(t) = \sum_{i} \sum_{k=1}^{K} a_k d_k(i) s_k(t - iT) \exp(j\theta_k(t)) + n(t), \tag{1}$$

where T is the signalling interval, and for the k-th user, a_k, $d_k(i)$, $s_k(t)$, and $\theta_k(t)$ are respectively the amplitude, the i-th differential data bit, the spreading signal, and the phase.

The relation between the differential data bits and the actual data bits, $\{b_k(i)\}$, may be expressed as:

$$d_k(i) = b_k(i)\, d_k(i-1). \tag{2}$$

Both the differential data bits and actual data bits are elements of $\{-1, +1\}$. The complex noise envelope, n(t), is assumed to be additive white Gaussian with variance, σ^2.

The real-valued spreading signals are non-zero only over [0, T] and are, without any loss of generality, assumed to be unit energy, i.e.:

$$\int_0^T s_k^2(t)dt = 1, \tag{3}$$

for $k \in \{1, ..., K\}$. The cross-correlation between the K users' signals are characterized by a $K \times K$ matrix **R** whose elements are defined by:

$$R_{kl} = \int_0^T s_k(t)s_l(t)dt, \tag{4}$$

for $k, l \in \{1...K\}$.

Two-stage Detector

The proposed two-stage detector is illustrated in Figure 1. The first stage, consisting of all processing through the tentative bit decisions, $\widetilde{\mathbf{b}}(i) = [\widetilde{b}_1(i), \widetilde{b}_2(i), \cdots, \widetilde{b}_K]^T$, is the bilinear decorrelating detector described in [5]. The vector of matched filter outputs, $\mathbf{z}(i) = [z_1(i), z_2(i), \cdots, z_K(i)]^T$, may be expressed as:

$$\mathbf{z}(i) = \mathbf{R}\,\mathbf{A}\,\mathbf{d}(i) + \mathbf{w}(i), \tag{5}$$

where $\mathbf{A} = \mathrm{diag}\{\, a_1e^{j\theta_1(i)}, a_2e^{j\theta_2(i)}, \ldots, a_Ke^{j\theta_K(i)} \,\}$ is the matrix with diagonal elements equal to the complex amplitudes of the K users' signals, $\mathbf{d}(i) = [d_1(i), d_2(i), \cdots, d_K(i)]^T$ is the vector of differential data bits, and $\mathbf{w}(i) = [w_1(i), w_2(i), \cdots, w_K(i)]^T$ is the vector of noise components of the matched filter outputs. The noise covariance matrix, $\mathbf{R}_w(j) = \frac{1}{2} E[\mathbf{w}(i)\, \mathbf{w}(i-j)^H]$, is given in terms of the signal correlation matrix by:

$$\mathbf{R}_w(j) = \sigma^2 \mathbf{R}\, \delta_j, \tag{6}$$

where δ_j is used to denote the Kronecker delta function.

The bilinear decorrelator premultiplies the matched filter outputs by the inverse of the signal cross-correlation matrix yielding:

$$\begin{aligned} \mathbf{y}(i) &= \mathbf{R}^{-1}\, \mathbf{z}(i) \\ &= \mathbf{A}\, \mathbf{d}(i) + \mathbf{v}(i), \end{aligned} \tag{7}$$

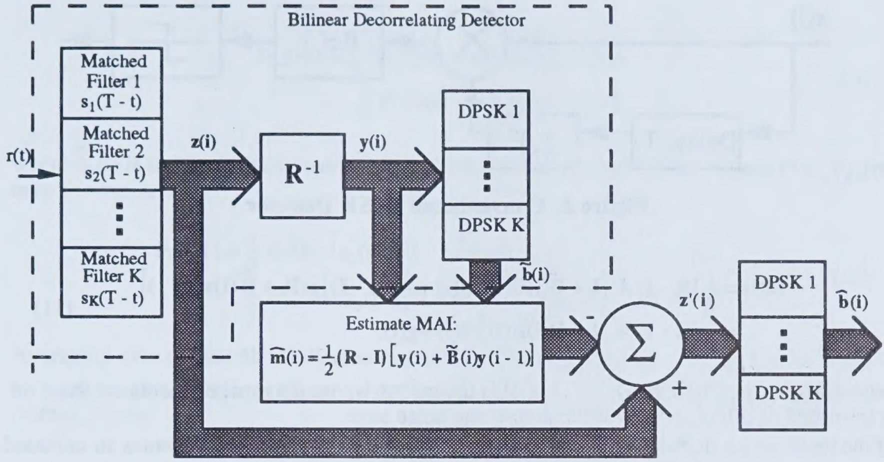

Bilinear Decorrelating Detector
Matched Filter 1 $s_1(T - t)$
Matched Filter 2 $s_2(T - t)$
Matched Filter K $s_K(T - t)$
r(t)
z(i)
$\mathbf{R}^{-1}$
y(i)
DPSK 1
DPSK K
$\tilde{b}$(i)
Estimate MAI:
$\widehat{m}(i) = \frac{1}{2}(\mathbf{R} - \mathbf{I})\left[\mathbf{y}(i) + \widetilde{\mathbf{B}}(i)\mathbf{y}(i - 1)\right]$
-
+
Σ
z'(i)
DPSK 1
DPSK K
$\widehat{b}$(i)

Figure 1. Two-Stage Noncoherent Multiuser Detector

where $\mathbf{v}(i) = \mathbf{R}^{-1}\,\mathbf{w}(i)$ is the noise term affecting the decorrelated matched filter outputs. The covariance matrix of $\mathbf{v}(i)$ is given by:

$$R_v(j) = \sigma^2\mathbf{R}^{-1}\,\delta_j. \tag{8}$$

Assuming that $\mathbf{R}$ is positive definite, it follows that although the premultiplication by $\mathbf{R}^{-1}$ results in the complete elimination of MAI, it also enhances the noise. The vector of decorrelated statistics, $\mathbf{y}(i)$, are then used to form the tentative bit decisions, $\tilde{\mathbf{b}}(i)$, using a bank of conventional DPSK detectors. A single such detector is shown in Figure 2. With $y_k(i)$ as an input, each DPSK detector outputs $\tilde{b}_k(i) = \operatorname{sgn}\{\operatorname{Re}[y_k(i)y_k(i-1)^*]\}$.

The two-stage detector uses the tentative bit decisions, $\tilde{\mathbf{b}}(i)$, and decorrelated matched filter outputs, $\mathbf{y}(i)$, to estimate the MAI components and subsequently subtract these estimates from the original matched filter outputs. To see how this is accomplished, it is useful to rewrite the vector of matched filter outputs as follows:

$$\mathbf{z}(i) = \mathbf{A}\,\mathbf{d}(i) + (\mathbf{R} - \mathbf{I})\,\mathbf{A}\,\mathbf{d}(i) + \mathbf{w}(i), \tag{9}$$

from which it can be readily seen that the matched filter outputs consist of the sum of a desired signal, MAI, and noise (first, second, and third terms of the RHS of (9) respectively). Let $\mathbf{m}(i) = (\mathbf{R} - \mathbf{I})\,\mathbf{A}\,\mathbf{d}(i)$ denote the MAI component. The proposed two-stage detector estimates $\mathbf{m}(i)$ using:

$$\widehat{\mathbf{m}}(i) = \frac{1}{2}(\mathbf{R} - \mathbf{I})\left[\mathbf{y}(i) + \widetilde{\mathbf{B}}(i)\mathbf{y}(i - 1)\right], \tag{10}$$

where $\widetilde{\mathbf{B}}(i) = \operatorname{diag}\{\tilde{\mathbf{b}}(i)\}$, and $\mathbf{I}$ is the $K \times K$ identity matrix. Note that through the substitution of (2) and (7), (10) may be reexpressed as:

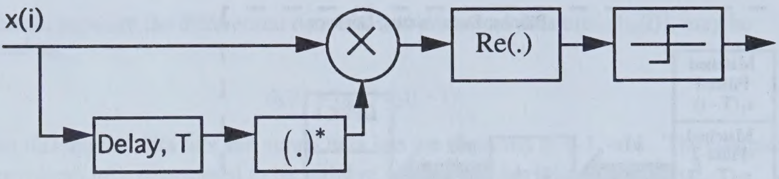


Figure 2. Conventional DPSK Detector

$$\begin{aligned}\widehat{\mathbf{m}}(i) &= \frac{1}{2}(\mathbf{R} - \mathbf{I})\mathbf{A}\left[\mathbf{I} + \widetilde{\mathbf{B}}(i)\mathbf{B}(i)\right]\mathbf{d}(i) + \frac{1}{2}(\mathbf{R} - \mathbf{I})\left[\mathbf{v}(i) + \widetilde{\mathbf{B}}(i)\mathbf{v}(i-1)\right] \\ &= \frac{1}{2}(\mathbf{R} - \mathbf{I})\mathbf{A}\left[\mathbf{I} + \widetilde{\mathbf{B}}(i)\mathbf{B}(i)\right]\mathbf{d}(i) + \mathbf{g}(i),\end{aligned} \tag{11}$$

where $\mathbf{B}(i) = \mathrm{diag}\{b_1(i), b_2(i), \cdots, b_K(i)\}$ is the matrix whose diagonal elements are the transmitted data bits and $\mathbf{g}(i)$ is the zero-mean noise term.
If the tentative bit decisions for all interfering users are all correct, $\widehat{\mathbf{m}}(i)$ becomes an unbiased estimator of the MAI, i.e., $\mathbf{E}\left[\widehat{\mathbf{m}}(i)\right] = \mathbf{m}(i)$.

After the MAI estimate is subtracted from the original matched filter outputs, the corrected outputs, $\mathbf{z}'(i) = \mathbf{z}(i) - \widehat{\mathbf{m}}(i)$, are then used to form final bit decisions using another bank of conventional DPSK detectors. The elements of $\mathbf{z}'(i)$ may be expressed as:

$$z'_k(i) = a_k e^{j\theta_k} d_k(i) + \frac{1}{2}\sum_{\substack{l=1 \\ l \neq k}}^{K} R_{kl} a_l e^{j\theta_l}\left[1 - \widetilde{b}_l(i) b_l(i)\right] d_k(i) + w_k(i) - g_k(i), \tag{12}$$

where:

$$g_k(i) = \frac{1}{2}\sum_{\substack{l=1 \\ l \neq k}}^{K} R_{kl}\left[v_l(i) + \widetilde{b}_l(i)\, v_l(i-1)\right]. \tag{13}$$

The final bit decision for user k may be expressed as:

$$\widehat{b}_k(i) = \mathrm{sgn}\left\{\mathrm{Re}\left[z'_k(i) z'_k(i-1)^*\right]\right\}. \tag{14}$$

Importantly, as can be seen from (14) and (12), each final bit decision made by the two-stage detector relies on matched filter outputs from three symboling intervals. The efficacious performance of the detector relies on the assumption that the received phase of each of the K users is constant over at least three signalling intervals, i.e., $\theta_k(i) = \theta_k(i-1) = \theta_k(i-2) \equiv \theta_k$. This assumption is a slight extension of that of the differentially coherent channel for which it is assumed the phase is constant over only two signalling intervals. The extension of the constant phase assumption over multiple signaling intervals is reasonable for many applications and has been made previously in numerous improvements to the conventional single-user DPSK detector [7-9].

Bit Error Probability

Let the bit error probability of the i-th bit of the k-th user be denoted $P_{k,i}(\sigma)$. This probability may be expressed as:

$$P_{k,i}(\sigma) = \frac{1}{2}\Pr\left[\widehat{b}_k(i) = +1 \mid b_k(i) = -1\right] + \frac{1}{2}\Pr\left[\widehat{b}_k(i) = -1 \mid b_k(i) = +1\right], \tag{15}$$

where $\widehat{b}_k(i)$ is the final bit estimate of the i-th bit of the k-th user. Using (2) and (14), $P_{k,i}(\sigma)$ may be rewritten as:

$$P_{k,i}(\sigma) = \frac{1}{2}\Pr\left[\mathrm{Re}\left(z_k'(i)\, z_k'(i-1)^*\right) > 0 \mid d_k(i)\, d_k(i-1) = -1\right] + \frac{1}{2}\Pr\left[\mathrm{Re}\left(z_k'(i)\, z_k'(i-1)^*\right) < 0 \mid d_k(i)\, d_k(i-1) = +1\right]. \tag{16}$$

Averaging over $\mathbf{d}_k$, the 3K - 1 element vector of differential data bits affecting the i-th final bit estimate of the k-th user, equal to $\left[\mathbf{d}(i)^T, \mathbf{d}(i-1)^T, \mathbf{d}(i-2)^T\right]^T$ with the entry for $d_k(i-2)$ deleted, yields:

$$P_{k,i}(\sigma) = \frac{1}{2^{(3K-1)}} \sum_{\mathbf{d}_k \in D^-} \Pr\left[D_k(i) > 0 \mid \mathbf{d}_k\right] + \frac{1}{2^{(3K-1)}} \sum_{\mathbf{d}_k \in D^+} \Pr\left[D_k(i) < 0 \mid \mathbf{d}_k\right], \tag{17}$$

where $D_k(i) = \mathrm{Re}\left(z_k'(i)\, z_k'(i-1)^*\right)$ has been introduced to shorten the notation and $D^{\pm} = \left\{\mathbf{d}_k\colon d_k(i)\, d_k(i-1) = \pm 1\right\}$.

Let $\mathbf{v}_k$ be the 3(K-1) element vector of noise terms affecting the decorrelated matched filter outputs given by $\left[\mathbf{v}(i)^T, \mathbf{v}(i-1)^T, \mathbf{v}(i-2)^T\right]^T$ with the entries corresponding to the k-th user deleted. Importantly, $\mathbf{v}_k$ is independent of both $w_k(i)$ and $w_k(i-1)$ (a proof for the completely analagous coherent case may be found in [4]). The conditional probabilities $\Pr\left[D_k(i) < 0 \mid \mathbf{d}_k, \mathbf{v}_k\right]$ and $\Pr\left[D_k(i) > 0 \mid \mathbf{d}_k, \mathbf{v}_k\right]$ may be found using a well known result (e.g., see Appendix 4.B of [10]) as:

$$\begin{aligned} \Pr\left[D_k(i) < 0 \mid \mathbf{d}_k, \mathbf{v}_k\right] &= Q\left(\frac{a}{\sigma}, \frac{b}{\sigma}\right) - \frac{1}{2} I_0\left(\frac{ab}{\sigma^2}\right) \exp\left(-\frac{a^2+b^2}{2\sigma^2}\right) \\ \Pr\left[D_k(i) > 0 \mid \mathbf{d}_k, \mathbf{v}_k\right] &= Q\left(\frac{b}{\sigma}, \frac{a}{\sigma}\right) - \frac{1}{2} I_0\left(\frac{ab}{\sigma^2}\right) \exp\left(-\frac{a^2+b^2}{2\sigma^2}\right), \end{aligned} \tag{18}$$

where:

$$\left\{\begin{matrix} b \\ a \end{matrix}\right\} = \frac{1}{2}\left|\, \overline{z_k'(i)} \pm \overline{z_k'(i-1)} \,\right|\Bigg|_{\mathbf{d}_k, \mathbf{v}_k}. \tag{19}$$

Q(a,b) and $I_0(x)$ are the Marcum-Q function and the zeroeth-order modified Bessel function of the first kind, respectively. The Marcum-Q function is defined as:

$$Q(a, b) = \int_b^{\infty} x \exp\left(-\frac{a^2+b^2}{2}\right) I_0(ax)\, dx. \tag{20}$$

It should be noted that the averaging in (19) is with respect to $w_k(i)$ and $w_k(i - 1)$ only, since except for these mutually independent noise terms, $z_k'(i)$ and $z_k'(i - 1)$ are completely specified for a fixed $\mathbf{d}_k$ and $\mathbf{v}_k$.

The bit error probability of the two-stage detector may be numerically computed† by noting that (17) may be rewritten as:

$$P_{k,i}(\sigma) = \frac{1}{2^{(3K-1)}} \sum_{\mathbf{d}_k \in D^-} \int_{\mathbf{v}_k} \Pr\left[D_k(i) > 0 \mid \mathbf{d}_k, \mathbf{v}_k\right] p_{\mathbf{v}_k}(\mathbf{v}_k)\, d\mathbf{v}_k + \frac{1}{2^{(3K-1)}} \sum_{\mathbf{d}_k \in D^+} \int_{\mathbf{v}_k} \Pr\left[D_k(i) < 0 \mid \mathbf{d}_k, \mathbf{v}_k\right] p_{\mathbf{v}_k}(\mathbf{v}_k)\, d\mathbf{v}_k, \tag{21}$$

where $p_{\mathbf{v}_k}(\mathbf{v}_k)$ is the probability density function of $\mathbf{v}_k$, and the integration is over the 6(K - 1) variables comprising the real and imaginary components of $\mathbf{v}_k$. Using the covariance matrix of $\mathbf{v}(i)$ given in (8), $p_{\mathbf{v}_k}(\mathbf{v}_k)$ may be found to be:

$$p_{\mathbf{v}_k}(\mathbf{v}_k) = \frac{1}{\left(2\pi\sigma^2\right)^{[3(K-1)]} \det^3\left[\left(\mathbf{R}^{-1}\right)_k\right]} \exp\left(-\frac{1}{2\sigma^2} \sum_{j=0}^{2} \mathbf{v}_k(i-j)^H \left[\left(\mathbf{R}^{-1}\right)_k\right]^{-1} \mathbf{v}_k(i-j)\right), \tag{22}$$

where $\left[\left(\mathbf{R}^{-1}\right)_k\right]$ is the submatrix formed by deleting the k-th row and column of the inverse of $\mathbf{R}$, and $\mathbf{v}_k(i)$ is equal to $\mathbf{v}(i)$ with the entry corresponding to the k-th user deleted.

Numerical Examples

In this section, two numerical examples comparing the performance of the proposed two-stage detector to the bilinear decorrelator are presented. The first example is the simple case where only two users are present. The bit error performance of both detectors for the case where one user is much stronger than the other (a2/a1 = 14 dB) is shown in Figure 3 (a cross-correlation coefficient, R_{12}, of 0.8 was used). Results of both Monte-Carlo simulation and numerical integration using (21) are shown. Note that the two curves agree very well. For the parameters shown, the two-stage detector requires about 1.5 dB less SNR than the bilinear decorrelator to achieve a bit error probability of 10^{-3}. This performance advantage may be explained as follows. When the interfering signal is much stronger than the desired signal, the decorrelating detector yields tentative bit estimates of the strong user with an exceedingly low error rate. The two-stage detector uses this near-perfect knowledge of the interfering user to form a corrected detection statistic which is less noisy than that used in the bilinear decorrelator. The performance of the two-stage detector over a range of interfering signal strength is illustrated in Figure 4. Note, consistent with the above explanation, that the two-stage detector performs much better than the bilinear decorrelator when the interfering signal is much stronger than the desired signal. There is, however, a region where the bilinear decorrelator outperforms the two-stage detector. In this region, where the interfering signal is somewhat weaker than the desired signal, noise introduced in estimating the MAI becomes a significant factor.

As a second example, the set of four waveforms shown in Figure 5, used previously in [5] to evaluate the performance of the bilinear decorrelating detector, were selected. The bit error performances of the bilinear decorrelator and the two-stage detector are shown in Figures 6 and 7 for two sets of parameters. For Figure 6, the amplitudes of the first three users were set equal to each other and the amplitude of the fourth user was varied. For Figure 7, the

---

†Due to the number of required integrations, Monte-Carlo simulation may be a more efficient means to evaluate the bit error performance, especially for large values of K.

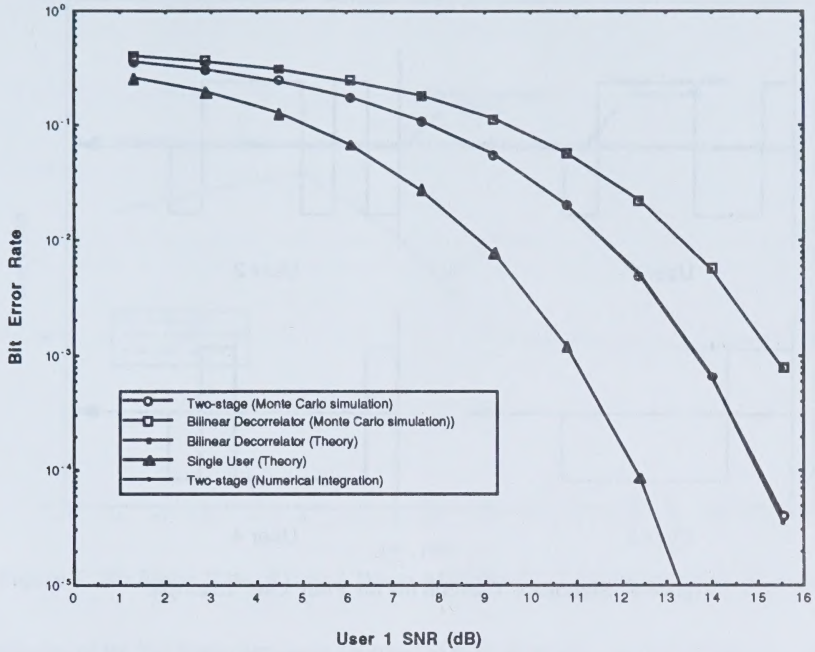


Figure 3. Bit Error Rates for Two User Example ($a_1 = 1, a_2 = 5$)

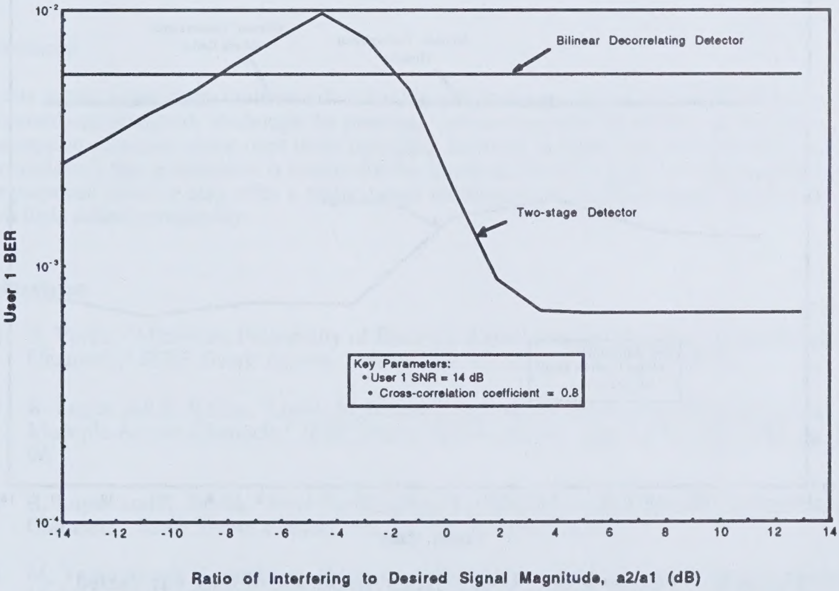


Figure 4. Two User Example: Bit Error Rate as a Function of the Ratio of Interfering to Desired Signal Magnitudes

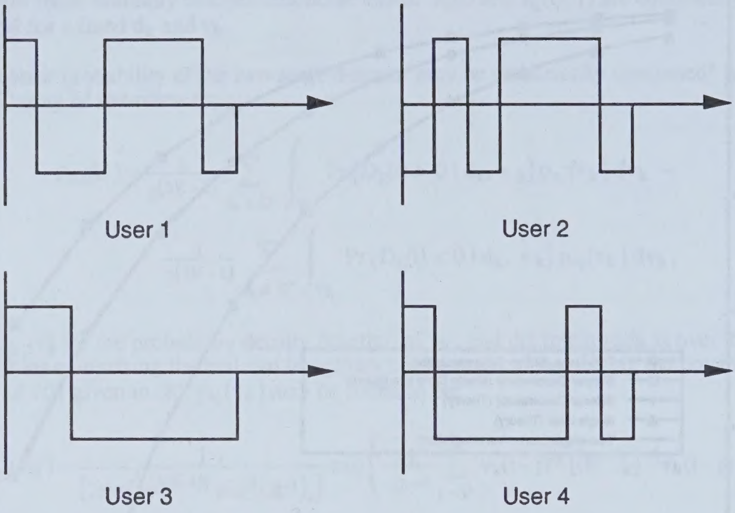

User 1
User 2
User 3
User 4

Figure 5. Signature Waveforms for Four User Example

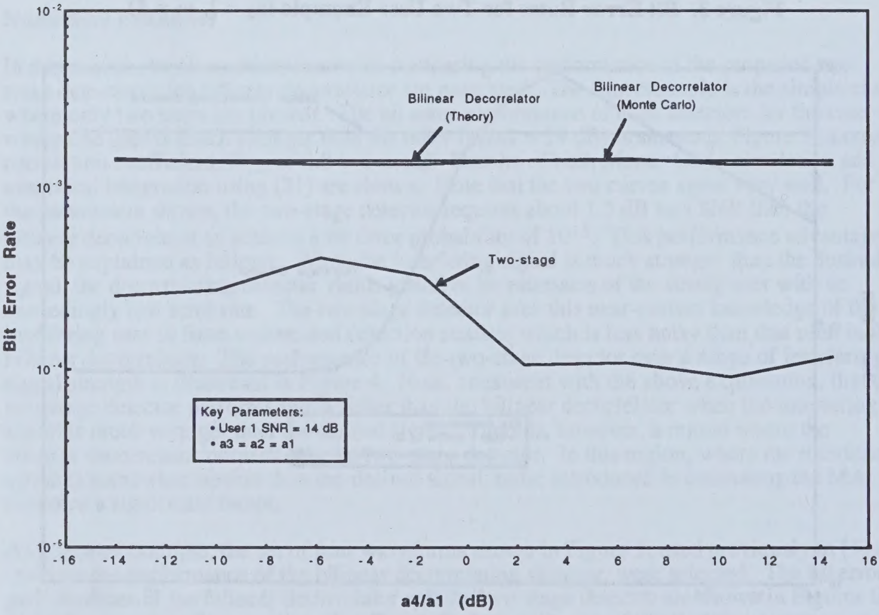


Figure 6: Bit Error Rate of User 1 When Amplitude of User 4 is Varied

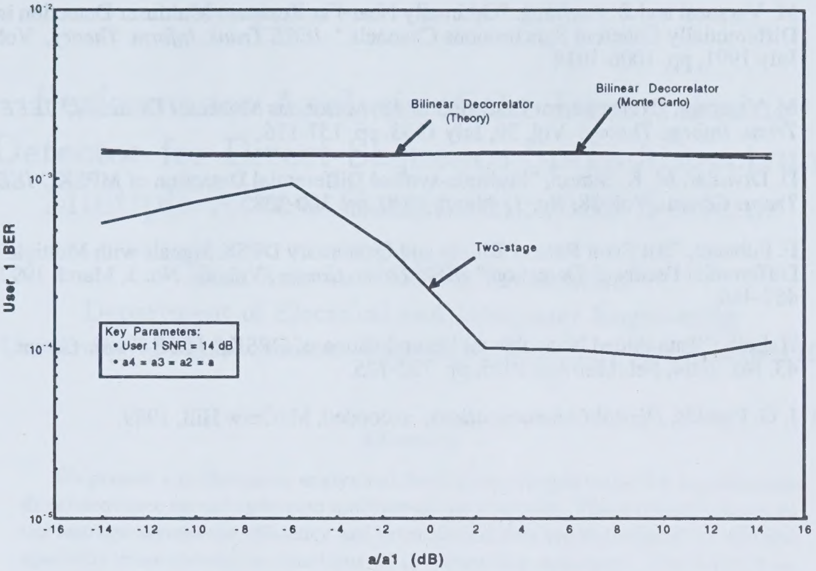


Figure 7. Bit Error Rate of User 1 When All Other Users' Amplitudes are Varied

amplitudes of the last three users were set equal to each other and varied while the first user's amplitude was held constant. For both sets of parameters, the two-stage detector outperforms the bilinear decorrelating detector.

Summary

In this paper, a two-stage multiuser detector for synchronous non-coherent CDMA has been proposed and analyzed. Although the proposed detector requires an additional channel assumption (constant phase over three signalling intervals as opposed to two for the bilinear decorrelator), this assumption is reasonable for many applications and for these applications, the proposed detector may offer a performance improvement over the bilinear decorrelator with little added complexity.

References

[1] S. Verdu, "Minimum Probability of Error for Asynchronous Gaussian Multiple-Access Channels," *IEEE Trans. Inform. Theory*, Vol. IT-32, Jan. 1986, pp. 85-96.

[2] R. Lupas and S. Verdu, "Linear Multiuser Detectors for Sychronous Code-Division Multiple Access Channels," *IEEE Trans. Inform. Theory*, Vol. IT-32, Jan. 1986, pp. 85-96.

[3] R. Lupas and S. Verdu, "Near-Far Resistance of Multiuser Detectors in Asynchronous Channels," *IEEE Trans. Comm.*, Vol. 38, No. 3, March 1990.

[4] M. Varanasi and B. Aazhang, "Multistage Detection in Asynchronous Code-Division Multiple-Access Communications," *IEEE Trans. Comm.*, Vol. 38, No. 4, April 1990, pp. 509-519.

[5] M. Varanasi and B. Aazhang, "Optimally Near-Far Resistant Multiuser Detection in Differentially Coherent Synchronous Channels," *IEEE Trans. Inform. Theory,* Vol. 37, July 1991, pp. 1006-1018.

[6] M. Varanasi, "Noncoherent Detection in Asynchronous Multiuser Channels," *IEEE Trans. Inform. Theory,* Vol. 39, July 1993, pp. 157-176.

[7] D. Divsalar, M. K. Simon, "Multiple-symbol Differential Detection of MPSK," *IEEE Trans. Comm.*, Vol. 38, No. 3, March 1990, pp. 300-308.

[8] F. Edbauer, "Bit Error Rate of Binary and Quaternary DPSK Signals with Multiple Differential Feedback Detection," *IEEE Trans. Comm.*, Vol. 40, No. 3, March 1992, pp. 457-460.

[9] H. Leib, "Data-Aided Noncoherent Demodulation of DPSK," *IEEE Trans. Comm.*, Vol. 43, No. 2/3/4, Feb/Mar/Apr 1995, pp. 722-725.

[10] J. G. Proakis, *Digital Communications,* seconded, McGraw Hill, 1989.

Performance Analysis of the Decorrelating Detector for Direct-Sequence Spread-Spectrum Multiple-Access Communication Systems

İrfan Acar and Sawasd Tantaratana
Department of Electrical and Computer Engineering
University of Massachusetts, Box 35110 Amherst, MA 01003-5110

Abstract

We present a performance analysis of the decorrelating detector for asynchronous direct-sequence spread-spectrum multiple-access channels. The resulting bounds on the average asymptotic efficiency and error probability are in terms of the discrete aperiodic cross-correlation functions of the spreading sequences. For random sequences, these bounds reduce to simple expressions in terms of the number of users and the spreading sequence length.

1 Introduction

The conventional approach to multiuser demodulation in direct-sequence spread-spectrum multiple-access (DS/SSMA) systems is to demodulate each user's signal as if the other users were not present. Although not optimal, this approach has the advantage of simplicity and performs well under most circumstances provided that the cross-correlations of the signature waveforms are low for all possible relative time delays. However, as the number of users increases or as the relative powers of the interfering signals become large (near-far problem), its performance degrades severely. When the signatures of all users are known, as for example in a base station in cellular communication systems, demodulation schemes that are optimum against both the background noise and the multiple-access interference can be employed to improve the performance at the expense of increased receiver complexity. The maximum-likelihood multiuser detector consists of a bank of matched filters followed by a dynamic programming algorithm that has a prohibitive computational complexity, which grows exponentially in the number of users [6]. Although the maximum-likelihood multiuser detector overcomes the shortcomings of the conventional approach and achieves a performance very near that of single-user communications, recent attention has shifted to near-far resistant suboptimum detectors of lower complexity [1, 2, 5, 7], in search of an appropriate tradeoff between complexity and performance.

Among the suboptimal multiuser detectors, the optimum linear detector for worst-case energies (decorrelating detector) achieves the near-far resistance of optimum multiuser detection, yet its complexity is only linear in the number of users [2]. Moreover, the decorrelating detector does not require knowledge of the received signal energies; thus, it becomes an attractive and reliable compromise between the conventional detector and the optimum multiuser detector. In [2], the error probability of the decorrelating detector was shown to be a function of the diagonal elements of the inverse of the signal cross-correlation matrix. The purpose of this paper is to further explore the performance of the decorrelating detector for coherent detection in DS/SSMA systems.

In the next section we will review the decorrelating detector in conjunction with DS/SSMA communications. In Section 3, we will obtain bounds on the performance of the decorrelating detector in terms of the discrete aperiodic cross-correlation functions of the signature sequences. In addition, for random sequences, the performance degradation due to multiple-access interference will be expressed in terms of the basic system parameters. Finally, we will conclude with some numerical results.

2 The Decorrelating Detector

Consider an asynchronous Gaussian multiple-access channel with K active users, each modulating antipodally a signature waveform, $a_k(t), k = 1, 2, \cdots, K$, $0 \le t \le T$, where T is the symbol interval. The received signal $r(t)$ can then be modeled as follows:

$$r(t) = \sum_{i=-M}^{M} \sum_{k=1}^{K} \sqrt{2\mathcal{E}_k(i)/T}\, b_k(i) a_k(t - iT - \tau_k) \cos(\omega_c t + \theta_k) + n(t), \tag{1}$$

where $2M + 1$ is the demodulation block length, $\mathcal{E}_k(i) > 0$ is the received energy of the k-th user's i-th bit, $b_k(i) \in \{-1, +1\}$ is the i-th binary data symbol of the k-th user, $\tau_k \in [0, T)$ is the relative delay of the k-th user, ω_c is the common carrier frequency, θ_k is the carrier phase of the k-th user, and $n(t)$ is the additive white Gaussian noise with two-sided power spectral density $N_0/2$. The signature waveform $a_k(t)$ assigned to the k-th user is given by

$$a_k(t) = \sum_{j=0}^{N-1} a_j^{(k)} \psi(t - jT_c), \quad 0 \le t \le T \tag{2}$$

where $a_0^{(k)}, a_1^{(k)}, \cdots, a_{N-1}^{(k)}$ is a signature sequence of length N. The chip waveform $\psi(t)$ is assumed to be a unit rectangular pulse of duration $T_c = T/N$.

It has been shown that the whole sequence of normalized matched filter outputs $y_m(l) = \sqrt{2/T} \int_{-\infty}^{\infty} r(t) a_m(t - lT - \tau_m) \cos(\omega_c t + \theta_m)\, dt$, for $m = 1, 2, \cdots, K$, and $l = -M, 1 - M, \cdots, M$ is a sufficient statistic for the decision on the most likely transmitted information sequence [6]. These matched filter outputs can be written in vector form as

$$\boldsymbol{y} = \boldsymbol{\mathcal{H}} \boldsymbol{\mathcal{E}} \boldsymbol{b} + \boldsymbol{n}, \tag{3}$$

where $\boldsymbol{\mathcal{H}}$ is a $(2M+1)K \times (2M+1)K$ symmetric block-Toeplitz cross-correlation matrix with the $(k, i), (m, l)$-th element[1] equal to $\mathcal{H}_{(k,i),(m,l)} = \frac{2}{T} \int_{-\infty}^{\infty} a_k(t - iT - \tau_k) a_m(t - lT - \tau_m) \cos(\omega_c t + \theta_k) \cos(\omega_c t + \theta_m)\, dt$, $\boldsymbol{\mathcal{E}}$ is a block-diagonal energy matrix with the $(k, i), (m, l)$-th element $\sqrt{\mathcal{E}_k(i)} \delta(i - l) \delta(k - m)$; and the vectors $\boldsymbol{y}, \boldsymbol{b}$, and $\boldsymbol{n}$ denote the matched filter output, information sequence, and noise vectors, respectively, with $y_m(l), b_m(l)$, and $n_m(l)$ being the m-th element of the l-th block, $l = -M, 1 - M, \cdots, M$, of the corresponding vector. Note that, since the signature waveforms are nonzero only for $0 \le t \le T$, we have

---

[1] Following the notation in [2], the $(k, i), (m, l)$-th element of a block matrix denotes the km-th element of the il-th block, with the blocks numbered from $-M$ to M.

$\mathcal{H}_{(k,i),(m,l)} = 0$ for $|i-l| > 1$. Assuming that $\omega_c T \gg 1$, $\mathcal{H}_{(k,i),(m,l)}$ for $i-1 \le l \le i+1$ can be expressed as

$$\mathcal{H}_{(k,i),(m,i)} = \begin{cases} \cos(\theta_m - \theta_k)\, \hat{R}_{mk}(\tau_m - \tau_k), & \tau_m \ge \tau_k, \\ \cos(\theta_m - \theta_k)\, \hat{R}_{km}(\tau_k - \tau_m), & \tau_m < \tau_k, \end{cases} \tag{4}$$

$$\mathcal{H}_{(k,i),(m,i-1)} = \mathcal{H}_{(m,i),(k,i+1)} = \begin{cases} \cos(\theta_m - \theta_k)\, R_{mk}(\tau_m - \tau_k), & \tau_m \ge \tau_k, \\ 0, & \tau_m < \tau_k, \end{cases} \tag{5}$$

where the normalized baseband correlation functions $R_{mk}(\tau)$ and $\hat{R}_{mk}(\tau)$ are defined for $0 \le \tau \le T$ as

$$R_{mk}(\tau) = \frac{1}{T} \int_0^{\tau} a_m(t + T - \tau)\, a_k(t) dt, \tag{6}$$

$$\hat{R}_{mk}(\tau) = \frac{1}{T} \int_{\tau}^{T} a_m(t - \tau)\, a_k(t) dt. \tag{7}$$

For $0 \le lT_c \le \tau \le (l+1)T_c \le T$, these two continuous-time correlation functions can be written as [3]

$$R_{mk}(\tau) = \frac{(l+1)T_c - \tau}{NT_c} C_{mk}(l - N) + \frac{\tau - lT_c}{NT_c} C_{mk}(l + 1 - N), \tag{8}$$

$$\hat{R}_{mk}(\tau) = \frac{(l+1)T_c - \tau}{NT_c} C_{mk}(l) + \frac{\tau - lT_c}{NT_c} C_{mk}(l + 1), \tag{9}$$

where the discrete aperiodic cross-correlation function $C_{mk}(l)$ for the signature sequences of the m-th and k-th users is defined as

$$C_{mk}(l) = \begin{cases} \sum_{j=0}^{N-1-l} a_j^{(m)} a_{j+l}^{(k)}, & 0 \le l \le N-1, \\ \sum_{j=0}^{N-1+l} a_{j-l}^{(m)} a_j^{(k)}, & 1 - N \le l < 0, \\ 0, & |l| \ge N. \end{cases} \tag{10}$$

When the signal energies are not known by the receiver, the maximum likelihood detector makes a decision that maximizes the maximum of the likelihood ratio over the unknown parameters. This yields the following estimate for the data:

$$\hat{\boldsymbol{b}} = \operatorname{sgn}\left[\boldsymbol{\mathcal{H}}^{-1} \boldsymbol{y}\right], \tag{11}$$

where sgn[$\cdot$] is $+1$ if the argument is positive and -1 if the argument is negative. This detector is known as the decorrelating detector [2]. A block diagram is depicted in Figure 1. The existence of $\boldsymbol{\mathcal{H}}^{-1}$ is guaranteed almost surely by the independence of the random delays $\tau_1, \tau_2, \cdots, \tau_K$ [2]. For the i-th bit of the k-th user, the estimate can be written as

$$\hat{b}_k(i) = \operatorname{sgn}\left[\mathcal{E}_k(i) b_k(i) + n'_k(i)\right], \tag{12}$$

where $n'_k(i)$ is a zero-mean Gaussian random variable with variance $\frac{N_0}{2}\mathcal{H}^{-1}_{(k,i),(k,i)}$. Therefore, for equally likely transmitted symbols, the probability of error in demodulating $b_k(i)$ is given by [2]

$$P_{k,i} = \frac{1}{2} \operatorname{erfc} \sqrt{\frac{\mathcal{E}_k(i)}{N_0 \mathcal{H}^{-1}_{(k,i),(k,i)}}}. \tag{13}$$

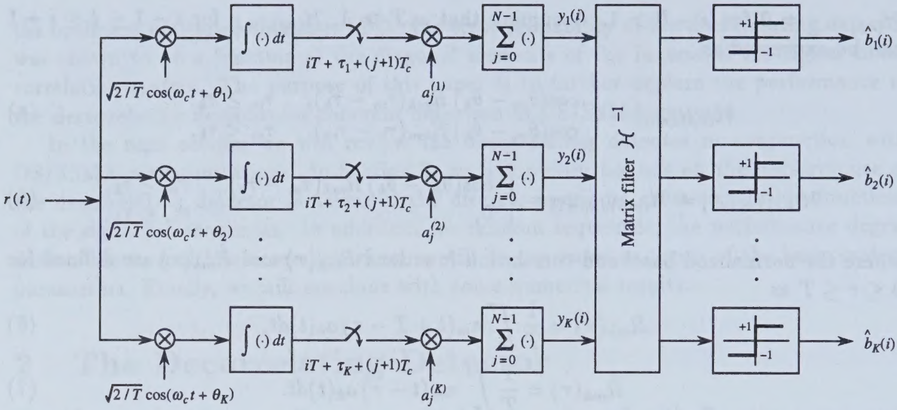


Figure 1: Decorrelating detector for DS/SSMA communications.

Note that the probability of error in (13) is independent of the energies of other users. Therefore, the decorrelating detector does not become multiple-access limited, no matter how strong the multiple-access interference is [2]. Moreover, since the decorrelating detector is optimal with respect to the worst possible distribution of energies, it has the same degree of near-far resistance as the maximum-likelihood multiuser detector [2]. In order to gain more insight on the performance of the decorrelating detector, we would like to know its average error probability. Note that the probability of error depends on the diagonal elements of $\mathcal{H}^{-1}$, and thus is a function of the random delays τ_k and phases θ_k. Since the nonlinearity of (13) prohibits a direct averaging, we will resort to bounds.

3 Bounds on the Average Asymptotic Efficiency

One measure used to show the performance degradation of multiuser detectors due to multiple-access interference is the asymptotic efficiency, η, defined as the ratio of the effective and actual energies [2, 6]. For the decorrelating detector, $\eta_{k,i}(\boldsymbol{\theta},\boldsymbol{\tau}) = 1/\mathcal{H}^{-1}_{(k,i),(k,i)}$, where $\boldsymbol{\theta} = [\theta_1, \theta_2, \cdots, \theta_K]^T$ and $\boldsymbol{\tau} = [\tau_1, \tau_2, \cdots, \tau_K]^T$. In this section, we will obtain upper and lower bounds on $E_{\boldsymbol{\theta},\boldsymbol{\tau}}(\eta_{k,i}(\boldsymbol{\theta},\boldsymbol{\tau}))$ by modeling the elements of $\boldsymbol{\theta}$ and $\boldsymbol{\tau}$ as mutually independent and uniformly distributed random variables. For this purpose, we define a matrix $\mathcal{R}$ obtained by deleting the (k,i)-th row[2] and (k,i)-th column of $\mathcal{H}$, and a vector $\boldsymbol{p}$ equal to the (k,i)-th column of $\mathcal{H}$ without the (k,i)-th element $(1-M \le i \le M-1)$.

A. Upper Bound

Note that $\eta_{k,i}(\boldsymbol{\theta},\boldsymbol{\tau})$ can be expressed as a constrained minimization of the quadratic form $\boldsymbol{x}^T\mathcal{H}\boldsymbol{x}$ [2], which allows us to obtain the following upper bound for the asymptotic efficiency:

$$\eta_{k,i}(\boldsymbol{\theta},\boldsymbol{\tau}) \quad = \quad \min_{x_k(i)=1} \boldsymbol{x}^T\mathcal{H}\boldsymbol{x}$$

---

[2] (k,i)-th row (column) refers to the k-th row (column) within the i-th block in the vertical (horizontal) direction

$$= \min_{\boldsymbol{y}} \left\{ 1 + 2\boldsymbol{p}^T \boldsymbol{y} + \boldsymbol{y}^T \boldsymbol{\mathcal{R}} \boldsymbol{y} \right\} \le 1 - 2\boldsymbol{p}^T \boldsymbol{p} + \boldsymbol{p}^T \boldsymbol{\mathcal{R}} \boldsymbol{p}, \tag{14}$$

where the inequality was obtained by evaluating the minimization over $\boldsymbol{y}$ at $\boldsymbol{y} = -\boldsymbol{p}$. In the Appendix A we show that

$$E_{\boldsymbol{\theta},\boldsymbol{\tau}}(\boldsymbol{p}^T \boldsymbol{p}) = \frac{1}{6N^3} \sum_{\substack{j=1 \\ j \ne k}}^{K} r_{kj}, \tag{15}$$

and

$$E_{\boldsymbol{\theta},\boldsymbol{\tau}}(\boldsymbol{p}^T (\boldsymbol{\mathcal{R}} - \boldsymbol{\mathcal{I}}) \boldsymbol{p}) = \frac{1}{240N^5} \sum_{\substack{j=1 \\ j \ne k}}^{K-1} \sum_{\substack{l=j+1 \\ l \ne k}}^{K} q_{kjl}, \tag{16}$$

where $r_{kj} \triangleq 2\mu_{kj}(0) + \mu_{kj}(1)$, $q_{kjl} \triangleq 66\nu_{kjl}(0) + 26\nu_{kjl}(1) + 26\nu_{kjl}(-1) + \nu_{kjl}(2) + \nu_{kjl}(-2)$, with

$$\mu_{kj}(\ell) = \sum_{m=1-N}^{N-1} C_{kj}(m) C_{kj}(m+\ell), \tag{17}$$

$$\nu_{kjl}(\ell) = \sum_{m=1-N}^{N-1} \sum_{n=1-N}^{N-1} C_{jl}(m) C_{kj}(n) C_{kl}(n+m+\ell). \tag{18}$$

Using (14),(15), and (16), we obtain the following upper bound:

$$E_{\boldsymbol{\theta},\boldsymbol{\tau}}(\eta_{k,i}(\boldsymbol{\theta},\boldsymbol{\tau})) \le 1 - \frac{1}{6N^3} \sum_{\substack{j=1 \\ j \ne k}}^{K} r_{kj} + \frac{1}{240N^5} \sum_{\substack{j=1 \\ j \ne k}}^{K-1} \sum_{\substack{l=j+1 \\ l \ne k}}^{K} q_{kjl}. \tag{19}$$

Note that the correlation parameters $\mu_{kj}(\ell)$ are also used in the performance evaluation of the conventional detector. In Appendix A, we show that $\mu_{kj}(\ell)$ and $\nu_{kj}(\ell)$ can be computed from the autocorrelation functions alone.

If we further assume that the chip values $a_j^{(k)}$ of the signature sequences are mutually independent discrete random variables with equal probability to be $+1$ or -1, then one can show that $E_{\boldsymbol{a}}(\mu_{kj}(\ell)) = N^2\delta(\ell)$ and $E_{\boldsymbol{a}}(\nu_{kjl}(\ell)) = N^3\delta(\ell)$, where $\boldsymbol{a}$ indicates that the expectation is with respect to $a_i^{(j)}$, $i = 0, 1, \cdots, N-1$, and $j = 1, 2, \cdots, K$. Hence, from (19) we obtain

$$E_{\boldsymbol{\theta},\boldsymbol{\tau},\boldsymbol{a}}(\eta_{k,i}(\boldsymbol{\theta},\boldsymbol{\tau},\boldsymbol{a})) \le 1 - \frac{K-1}{3N} + \frac{11(K-1)(K-2)}{80N^2}. \tag{20}$$

By choosing $\boldsymbol{y} = -\zeta\boldsymbol{p}$, where ζ is an arbitrary parameter, instead of $\boldsymbol{y} = -\boldsymbol{p}$ in (14), an upper bound as a function of ζ can be obtained. It can be shown that this bound is minimized at $\zeta = \frac{80N}{80N + 33(K-2)}$, yielding

$$E_{\boldsymbol{\theta},\boldsymbol{\tau},\boldsymbol{a}}(\eta_{k,i}(\boldsymbol{\theta},\boldsymbol{\tau},\boldsymbol{a})) \le 1 - \frac{80(K-1)}{240N + 99(K-2)}, \tag{21}$$

which is tighter than (20).

B. Lower Bound

To obtain a lower bound on the average asymptotic efficiency we first note that $\eta_{k,i}(\boldsymbol{\theta},\boldsymbol{\tau}) = 1 - \boldsymbol{p}^T\boldsymbol{\mathcal{R}}^{-1}\boldsymbol{p}$. In Appendix B, we show that $E_{\boldsymbol{\theta},\boldsymbol{\tau}}(\boldsymbol{p}^T\boldsymbol{\mathcal{R}}^{-1}\boldsymbol{p}) \leq E_{\boldsymbol{\theta},\boldsymbol{\tau}}(\boldsymbol{p}^T\boldsymbol{p})$, when $E_{\boldsymbol{\theta},\boldsymbol{\tau}}(\boldsymbol{p}^T\boldsymbol{\mathcal{R}}\boldsymbol{p}) \geq E_{\boldsymbol{\theta},\boldsymbol{\tau}}(\boldsymbol{p}^T\boldsymbol{p})$. Therefore, under this condition, we have

$$\begin{aligned} E_{\boldsymbol{\theta},\boldsymbol{\tau}}(\eta_{k,i}(\boldsymbol{\theta},\boldsymbol{\tau})) &= 1 - E_{\boldsymbol{\theta},\boldsymbol{\tau}}(\boldsymbol{p}^T\boldsymbol{\mathcal{R}}^{-1}\boldsymbol{p}) \\ &\geq 1 - E_{\boldsymbol{\theta},\boldsymbol{\tau}}(\boldsymbol{p}^T\boldsymbol{p}) = 1 - \frac{1}{6N^3}\sum_{\substack{j=1\\ j\neq k}}^{K} r_{kj}. \end{aligned} \tag{22}$$

For random signature sequences, $E_{\boldsymbol{\theta},\boldsymbol{\tau},\boldsymbol{a}}(\boldsymbol{p}^T(\boldsymbol{\mathcal{R}}-\boldsymbol{\mathcal{I}})\boldsymbol{p}) = 11(K-1)(K-2)/(80N^2)$ is nonnegative. Therefore, $E_{\boldsymbol{\theta},\boldsymbol{\tau},\boldsymbol{a}}(\boldsymbol{p}^T\boldsymbol{\mathcal{R}}^{-1}\boldsymbol{p}) \leq E_{\boldsymbol{\theta},\boldsymbol{\tau},\boldsymbol{a}}(\boldsymbol{p}^T\boldsymbol{p})$. Hence, for random sequences, we have

$$1 - \frac{K-1}{3N} \leq E_{\boldsymbol{\theta},\boldsymbol{\tau},\boldsymbol{a}}(\eta_{k,i}(\boldsymbol{\theta},\boldsymbol{\tau},\boldsymbol{a})) \leq 1 - \frac{K-1}{3N + \frac{99}{80}(K-2)}. \tag{23}$$

When K is small compared to N, we can use the approximation $E_{\boldsymbol{\theta},\boldsymbol{\tau},\boldsymbol{a}}(\eta_{k,i}(\boldsymbol{\theta},\boldsymbol{\tau},\boldsymbol{a})) \approx 1 - \frac{K-1}{3N}$. This approximation and the bounds in (23) allow us to quantify the performance of the decorrelating detector in terms of the basic system parameters, and can be used for preliminary system design. Note that all of the bounds in this paper are independent of M. In particular, they hold for $M \to \infty$ as well.

Using the concavity of the complementary error function, we can obtain a corresponding upper bound on the average error probability of the decorrelating detector. From (13) and (22)

$$\frac{1}{2}\operatorname{erfc}\sqrt{\frac{\mathcal{E}_k(i)}{N_0}} \leq E_{\boldsymbol{\theta},\boldsymbol{\tau}}(P_{k,i}) \leq \frac{1}{2}\operatorname{erfc}\sqrt{\frac{\mathcal{E}_k(i)}{N_0}\left[1 - \frac{1}{6N^3}\sum_{\substack{j=1\\ j\neq k}}^{K} r_{kj}\right]}. \tag{24}$$

Note that the lower bound is simply the single user error probability. For random sequences,

$$\frac{1}{2}\operatorname{erfc}\sqrt{\frac{\mathcal{E}_k(i)}{N_0}} \leq E_{\boldsymbol{\theta},\boldsymbol{\tau},\boldsymbol{a}}(P_{k,i}) \leq \frac{1}{2}\operatorname{erfc}\sqrt{\frac{\mathcal{E}_k(i)}{N_0}\left(1 - \frac{K-1}{3N}\right)}. \tag{25}$$

Note that the bounds in (24) and (25) also hold for the error probability of the maximum-likelihood multiuser detector.

For comparison, the average error probability of a conventional detector for DS/SSMA systems with power control is given by [3]

$$E_{\boldsymbol{\theta},\boldsymbol{\tau},\boldsymbol{a}}(P_{k,i}^{\text{conv}}) \approx \frac{1}{2}\operatorname{erfc}\left[\frac{N_0}{\mathcal{E}_k(i)} + \frac{1}{3N^3}\sum_{\substack{j=1\\ j\neq k}}^{K} r_{kj}\right]^{-\frac{1}{2}}, \tag{26}$$

and for random sequences by

$$E_{\boldsymbol{\theta},\boldsymbol{\tau},\boldsymbol{a}}(P_{k,i}^{\text{conv}}) \approx \frac{1}{2}\operatorname{erfc}\left[\frac{N_0}{\mathcal{E}_k(i)} + \frac{2(K-1)}{3N}\right]^{-\frac{1}{2}}. \tag{27}$$

A quantitative comparison between the performances of the decorrelating detector and the conventional detector can be obtained easily from the expressions above.

In Figure 2, we plot the bounds given in (23) along with the simulation results for the average asymptotic efficiency for $M = 5$ and $i = 0$. We observe that the simulation results are close to the lower bound, justifying the approximation $E_{\boldsymbol{\theta},\boldsymbol{\tau},\boldsymbol{a}}(\eta_{k,i}(\boldsymbol{\theta},\boldsymbol{\tau})) \approx 1 - \frac{K-1}{3N}$.

In Figure 3 the error probability bounds of (25) for the decorrelating detector is plotted along with the simulation results for $N = 31$. Also, simulation results for the conventional detector are given for the case of equal energy interferers. We observe that the bound is very tight for low K/N. For high K/N, the bound is within 0.5 dB of the simulation results. Similar results have been obtained for different sequence lengths, but cannot be presented here due to limited space.

4 Conclusion

We have presented bounds on the average asymptotic efficiency and error probability of the decorrelating detector for asynchronous direct-sequence spread-spectrum multiple-access channels. The bounds are in terms of the discrete aperiodic cross-correlation functions, which are well known for their use in the performance evaluation of the conventional detector for DS/SSMA systems. These bounds reduce to simple expressions in terms of the number of users and the signature sequence length, when random sequences are employed. These bounds quantify the performance of the decorrelating detector in terms of the basic system parameters, thus providing more insight into the effect of multiple access interference on the performance of the decorrelating detector.

APPENDIX A

It can easily be shown that

$$\boldsymbol{p}^T\boldsymbol{p} = \sum_{\substack{j=1\\ j\neq k}}^{K} [\mathcal{H}^2_{(k,i),(j,i)} + \mathcal{H}^2_{(k,i),(j,i-1)} + \mathcal{H}^2_{(k,i),(j,i+1)}]. \tag{A.1}$$

Hence, from (4) and (5), and using the fact that $E_{\theta_j,\theta_k}(\cos^2(\theta_j - \theta_k)) = \frac{1}{2}$ we have

$$\begin{aligned}
E_{\boldsymbol{\theta},\boldsymbol{\tau}}(\boldsymbol{p}^T\boldsymbol{p}) &= \frac{1}{2T^2}\sum_{\substack{j=1\\ j\neq k}}^{K}\Bigg[\int_0^T\int_0^{\tau_j}\left[R^2_{jk}(\tau_j-\tau_k)+\hat{R}^2_{jk}(\tau_j-\tau_k)\right]d\tau_k\,d\tau_j \\
&\qquad + \int_0^T\int_{\tau_j}^T\left[R^2_{kj}(\tau_k-\tau_j)+\hat{R}^2_{kj}(\tau_k-\tau_j)\right]d\tau_k\,d\tau_j\Bigg] \\
&= \frac{1}{2T^2}\sum_{\substack{j=1\\ j\neq k}}^{K}\int_0^T\left[R^2_{jk}(\alpha)+\hat{R}^2_{jk}(\alpha)+R^2_{kj}(\alpha)+\hat{R}^2_{kj}(\alpha)\right](T-\alpha)\,d\alpha \\
&= \frac{1}{2T}\sum_{\substack{j=1\\ j\neq k}}^{K}\sum_{m=0}^{N-1}\int_{mT_c}^{(m+1)T_c}\left[R^2_{kj}(\alpha)+\hat{R}^2_{kj}(\alpha)\right]d\alpha \\
&= \frac{1}{6N^3}\sum_{\substack{j=1\\ j\neq k}}^{K}[2\mu_{kj}(0)+\mu_{kj}(1)],
\end{aligned} \tag{A.2}$$

where the cross-correlation parameters $\mu_{kj}(\ell)$ are defined in (17).

It can be shown that

$$\begin{aligned}
\boldsymbol{p}^T(\mathcal{R}-\mathcal{I})\boldsymbol{p} \;=\; 2\sum_{\substack{j=1\\ j\neq k}}^{K}\sum_{\substack{l=j+1\\ l\neq k}}^{K} &\Big[\mathcal{H}_{(j,i),(l,i)}\Big(\mathcal{H}_{(k,i),(j,i)}\mathcal{H}_{(k,i),(l,i)}+\mathcal{H}_{(k,i),(j,i-1)}\mathcal{H}_{(k,i),(l,i-1)}\\
&+\mathcal{H}_{(j,i),(k,i-1)}\mathcal{H}_{(l,i),(k,i-1)}\Big)+\mathcal{H}_{(j,i),(k,i)}\Big(\mathcal{H}_{(j,i),(l,i-1)}\mathcal{H}_{(k,i),(l,i-1)}\\
&+\mathcal{H}_{(l,i),(j,i-1)}\mathcal{H}_{(l,i),(k,i-1)}\Big)+\mathcal{H}_{(k,i),(l,i)}\Big(\mathcal{H}_{(l,i),(j,i-1)}\mathcal{H}_{(k,i),(j,i-1)}\\
&+\mathcal{H}_{(j,i),(l,i-1)}\mathcal{H}_{(j,i),(k,i-1)}\Big)\Big]. \qquad \text{(A.3)}
\end{aligned}$$

Hence, from (4) and (5), and using the fact that $E_{\theta_j,\theta_k,\theta_l}(\cos(\theta_l-\theta_j)\cos(\theta_j-\theta_k)\cos(\theta_l-\theta_k))=\frac{1}{4}$ we have

$$E_{\boldsymbol{\theta},\boldsymbol{\tau}}(\boldsymbol{p}^T(\boldsymbol{R}-\boldsymbol{I})\boldsymbol{p}) = \qquad \text{(A.4)}$$

$$\begin{aligned}
\frac{1}{2T^3}\sum_{\substack{j=1\\ j\neq k}}^{K-1}\sum_{\substack{l=j+1\\ l\neq k}}^{K}\int_0^T\!\!\int_0^T\!\!\int_0^T &\Big[\hat{R}_{i_1i_2}(\tau_{i_1}-\tau_{i_2})\hat{R}_{i_1i_3}(\tau_{i_1}-\tau_{i_3})\hat{R}_{i_2i_3}(\tau_{i_2}-\tau_{i_3})\\
&+\hat{R}_{i_1i_2}(\tau_{i_1}-\tau_{i_2})R_{i_1i_3}(\tau_{i_1}-\tau_{i_3})R_{i_2i_3}(\tau_{i_2}-\tau_{i_3})\\
&+R_{i_1i_2}(\tau_{i_1}-\tau_{i_2})R_{i_1i_3}(\tau_{i_1}-\tau_{i_3})\hat{R}_{i_2i_3}(\tau_{i_2}-\tau_{i_3})\Big]\,d\tau_k\,d\tau_j\,d\tau_l, \qquad \text{(A.5)}
\end{aligned}$$

where $i_n, n\in\{1,2,3\}$ stands for the index of the n-th element of $\{\tau_j,\tau_k,\tau_l\}$ arranged in descending order. For example, if $\tau_l\geq\tau_k\geq\tau_j$, then $i_1=l, i_2=k$, and $i_3=j$.

In order to reduce the triple integral in (A.5) to a double integral we note that

$$\begin{aligned}
&\int_0^T\!\!\int_0^{\tau_1}\!\!\int_0^{\tau_2}\Big[\hat{R}_{12}(\tau_1-\tau_2)\hat{R}_{13}(\tau_1-\tau_3)\hat{R}_{23}(\tau_2-\tau_3)+\hat{R}_{12}(\tau_1-\tau_2)R_{13}(\tau_1-\tau_3)R_{23}(\tau_2-\tau_3)\\
&\qquad+R_{12}(\tau_1-\tau_2)R_{13}(\tau_1-\tau_3)\hat{R}_{23}(\tau_2-\tau_3)\Big]\,d\tau_3\,d\tau_2\,d\tau_1\\
&=\int_0^T\!\!\int_0^{T-\alpha}\Big[\hat{R}_{12}(\alpha)\hat{R}_{13}(\alpha+\beta)\hat{R}_{23}(\beta)+\hat{R}_{12}(\alpha)R_{13}(\alpha+\beta)R_{23}(\beta)\\
&\qquad+R_{12}(\alpha)R_{13}(\alpha+\beta)\hat{R}_{23}(\beta)\Big](T-\alpha-\beta)\,d\beta\,d\alpha\\
&=\sum_{m=0}^{N-1}\sum_{n=m}^{N-1}\int_{mT_c}^{(m+1)T_c}\int_{(n-m)T_c}^{(n+1)T_c-\alpha}\Big[\hat{R}_{12}(\alpha)\hat{R}_{13}(\alpha+\beta)\hat{R}_{23}(\beta)+\hat{R}_{12}(\alpha)R_{13}(\alpha+\beta)R_{23}(\beta)\\
&\qquad+R_{12}(\alpha)R_{13}(\alpha+\beta)\hat{R}_{23}(\beta)\Big](T-\alpha-\beta)\,d\beta\,d\alpha.\\
&+\sum_{m=0}^{N-2}\sum_{n=m}^{N-2}\int_{mT_c}^{(m+1)T_c}\int_{(n+1)T_c-\alpha}^{(n-m+1)T_c}\Big[\hat{R}_{12}(\alpha)\hat{R}_{13}(\alpha+\beta)\hat{R}_{23}(\beta)+\hat{R}_{12}(\alpha)R_{13}(\alpha+\beta)R_{23}(\beta)\\
&\qquad+R_{12}(\alpha)R_{13}(\alpha+\beta)\hat{R}_{23}(\beta)\Big](T-\alpha-\beta)\,d\beta\,d\alpha. \qquad \text{(A.6)}
\end{aligned}$$

After we repeat this computation in (A.6) for all six permutations of j, k, and l, and combine the results, we arrive at (19).

In [4], it is shown that using the following property of the aperiodic crosscorrelation functions,

$$\sum_{j=1-N}^{N-1}C_{xy}(j)C_{xz}(j-i)=\sum_{j=1-N}^{N-1}C_{xx}(j)C_{yz}(j-i), \qquad \text{(A.7)}$$

$\mu_{kj}(\ell)$ can be written in terms of only the autocorrelation functions:

$$\mu_{kjl}(\ell) = \sum_{m=1-N}^{N-1} C_{kk}(m)C_{jj}(m+\ell). \tag{A.8}$$

In a similar fashion, using (A.7) twice, $\nu_{kjl}(\ell)$ can also be expressed in terms of only the autocorrelation functions:

$$\nu_{kjl}(\ell) = \sum_{m=1-N}^{N-1} \sum_{n=1-N}^{N-1} C_{kk}(m)C_{jj}(n)C_{ll}(n+m+\ell). \tag{A.9}$$

APPENDIX B

Let λ_i and $\boldsymbol{u}_i$ denote the eigenvalues and the eigenvectors of $\mathcal{R}$. Note that $\mathcal{R}$ is positive definite; therefore, its eigenvalues are positive. In particular, the largest eigenvalue, λ_{max}, is positive. Then,

$$\begin{aligned} E_{\boldsymbol{\theta},\boldsymbol{\tau}}(\boldsymbol{p}^T(\mathcal{I}-\mathcal{R}^{-1})\boldsymbol{p}) &= E_{\boldsymbol{\theta},\boldsymbol{\tau}}\left(\sum_i(\boldsymbol{p}^T\boldsymbol{u}_i)^2\left[1-\frac{1}{\lambda_i}\right]\right) \\ &\geq \frac{1}{\lambda_{max}}E_{\boldsymbol{\theta},\boldsymbol{\tau}}\left(\sum_i(\boldsymbol{p}^T\boldsymbol{u}_i)^2(\lambda_i-1)\right) \\ &= \frac{1}{\lambda_{max}}E_{\boldsymbol{\theta},\boldsymbol{\tau}}(\boldsymbol{p}^T(\mathcal{R}-\mathcal{I})\boldsymbol{p}). \end{aligned} \tag{B.1}$$

Thus, when $E_{\boldsymbol{\theta},\boldsymbol{\tau}}(\boldsymbol{p}^T\mathcal{R}\boldsymbol{p}) \geq E_{\boldsymbol{\theta},\boldsymbol{\tau}}(\boldsymbol{p}^T\boldsymbol{p})$, the right hand side is greater than or equal to zero and therefore $E_{\boldsymbol{\theta},\boldsymbol{\tau}}(\boldsymbol{p}^T\mathcal{R}^{-1}\boldsymbol{p}) \leq E_{\boldsymbol{\theta},\boldsymbol{\tau}}(\boldsymbol{p}^T\boldsymbol{p})$.

References

[1] A. Duel-Hallen, "A family of multiuser decision-feedback detectors for asynchronous code-division multiple-access channels," *IEEE Trans. Commun.*, vol. 43, pp. 421-434, Feb./Mar./Apr. 1995.

[2] R. Lupas and S. Verdú, "Near-far resistance of multiuser detectors in asynchronous channels," *IEEE Trans. Commun.*, vol. 38, pp. 496-508, Apr. 1990.

[3] M. B. Pursley, "Performance evaluation for phase-coded spread- spectrum multiple-access communication—Part I: System analysis," *IEEE Trans. Commun.*, vol. COM-25, pp. 795-799, Aug. 1977.

[4] M. B. Pursley and D. V. Sarwate , "Performance evaluation for phase-coded spread-spectrum multiple-access communication—Part II: Code sequence analysis," *IEEE Trans. Commun.*, vol. COM-25, pp. 800-803, Aug. 1977.

[5] M. K. Varanasi and B. Aazhang, "Multistage detection in asynchronous code-division multiple-access communications," *IEEE Trans. Commun.*, vol. 38, pp. 509-519, Apr. 1990.

[6] S. Verdú, "Minimum probability of error for asynchronous Gaussian multiple-access channels," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 85-96, Jan. 1986.

[7] Z. Xie, R. T. Short and C. K. Rushforth, "A family of suboptimum detectors for coherent multiuser communications," *IEEE JSAC*, pp. 683-690, May 1990.

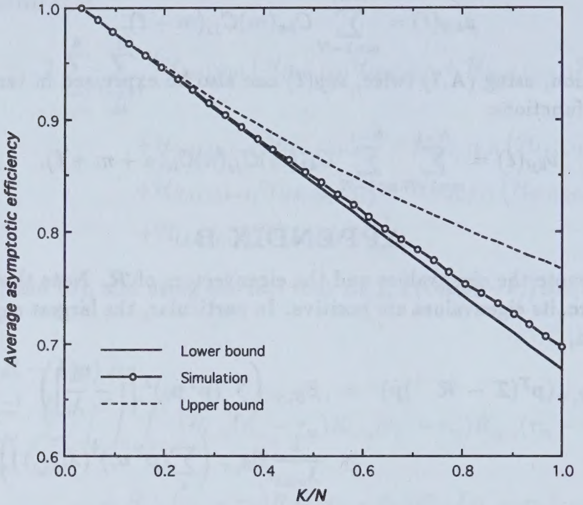


Figure 2: Average asymptotic efficiency of the decorrelating detector for random sequences of length $N = 31$.

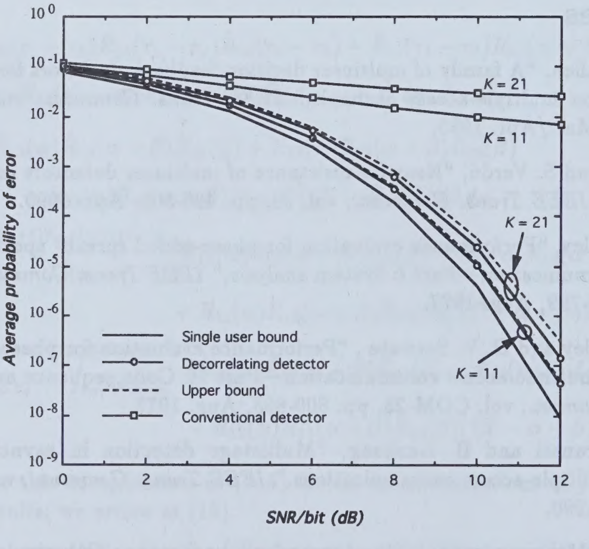


Figure 3: Average error probability of the decorrelating detector for random sequences of length $N = 31$.

A Closed-Loop Joint PN Phase and Carrier Phase Acquisition Scheme for DS/SS Systems

Murat Salih and **Sawasd Tantaratana**

Department of Electrical and Computer Engineering,
University of Massachusetts, Amherst, MA 01003-5110

Abstract

We propose a closed-loop system for joint acquisition of the carrier phase and the pseudo-noise (PN) signal phase in direct-sequence spread-spectrum (DS/SS) systems. The system has a voltage-controlled-clock (VCC) loop for PN phase acquisition and a voltage-controlled-oscillator (VCO) loop for carrier phase acquisition. These loops work together. The VCC loop uses the carrier phase estimate of the VCO loop to form its code phase estimate. Similary, the VCO loop uses the code phase estimate of the VCC loop to form its carrier phase estimate. Both loops use an auxiliary signal, as opposed to the PN signal itself, for correlation with the incoming signal. The correlation function of the auxiliary signal and the PN signal has a triangle shape that covers essentially the entire period of the PN signal. Consequently, the correlator outputs in both loops provide the directions for the local code phase and the carrier phase updates. Mean and variance of the code phase acquisition time are derived and are compared with those of the conventional coherent and noncoherent serial-search receivers, and the coherent closed-loop code phase acquisition receiver. Results suggest that, the proposed system performs better than both the noncoherent and the coherent serial-search systems, and almost as good as the coherent closed-loop system.

1 Introduction

An important task of a direct-sequence spread-spectrum (DS/SS) receiver is to generate a local replica of the transmitted sequence and to synchronize its phase to that of the incoming sequence. This synchronization process is typically accomplished in two stages: acquisition and tracking. In the first stage (acquisition), the phases of the two sequences are aligned to within a small range of one chip or less. Then in the second stage the tracking circuitry makes the fine adjustment and maintains the two codes in synchronism by means of a closed loop operation. Many types of DS/SS acquisition and tracking systems have been reported and analyzed [1].

There are a variety of acquisition receivers depending on the type of the detector and the search algorithm [1]. If the carrier phase is available during acquisition, then coherent pseudo-noise (PN) acquisition systems are used. If the carrier phase is not available during acquisition, then noncoherent PN acquisition systems are used. As an alternative to the noncoherent systems, a joint PN phase and carrier phase acquisition system can be used and that is the subject of this paper.

A practical serial implementation of the noncoherent maximum-likelihood (ML) algorithm is shown in Figure 1 . The code phase uncertainty region is quantized into a number of discrete phases. A correlator is used to examine each discrete phase for a fixed duration of $n_1 T_c$ seconds until a phase alignment is declared. When phase alignment is rejected, the phase is updated to the next one in a unidirectional fashion from one end of the uncertainty region to the other.

Although joint PN phase and carrier phase tracking has been reported [2], [3], joint PN phase and carrier phase acquisition has not been investigated to our knowledge. In this paper, we propose a closed-loop joint PN phase and carrier phase acquisition scheme for DS/SS systems. The system provides the directions for the local code phase and the carrier phase adjustments. This is accomplished by using an auxiliary signal, as opposed to the PN signal itself, to correlate with the incoming signal.

2 The Proposed Receiver

The baseband representation of a received DS/SS signal during acquisition is as follows:

$$u(t) = \sqrt{2P}c(t-\tau)e^{j\phi} + z(t) \tag{1}$$

where P is the average signal power assumed to be known, τ and ϕ are the unknown code phase and the carrier phase, respectively, to be estimated by a joint acquisition system, $z(t)$ is the zero mean AWGN with autocorrelation function $E\{z(t+\tau)z^*(t)\} = 2N_o\delta(\tau)$, and $c(t)$ is the PN code waveform expressed as

$$c(t) = \sum_{k=-\infty}^{\infty} c_k P_{T_c}(t-kT_c) \tag{2}$$

In this expression, c_k is the k^{th} chip of a PN sequence which is periodic with period N, i.e., $c_k = c_{k+N}$ for all k, and $P_{T_c}(.)$ is the unit amplitude rectangular pulse shape in the interval $[0,T_c]$, where T_c is the chip duration. We assume that $\{c_k\}$ is an m-sequence and that there is no data modulation during acquisition. Only single-user case is considered here.

We propose the receiver shown in Figure 2. The auxiliary signal $\alpha(t)$ is defined as follows:

$$\begin{aligned} \alpha(t) &= \sum_{m=-\frac{N-3}{2}}^{\frac{N-3}{2}} \left[\tfrac{N-1}{2} - |m|\right] c(t-mT_c) & (3) \\ &= \sum_{k=-\infty}^{\infty} \alpha_k P_{T_c}(t-kT_c) & (4) \end{aligned}$$

where,

$$\alpha_k = \sum_{m=-\frac{N-3}{2}}^{\frac{N-3}{2}} \left[\tfrac{N-1}{2} - |m|\right] c_{k-m} \tag{5}$$

The sequence $\{\alpha_k\}$ and the waveform $\alpha(t)$ are periodic with the respective periods N and NT_c, which are the same as the periods of $\{c_k\}$ and $c(t)$, respectively. As seen from (3), $\alpha(t)$ is a weighted sum of shifted versions of the m-sequence waveform $c(t)$. We define the periodic cross-correlation function between $c(t)$ and $\alpha(t)$ as follows:

$$\begin{aligned} R_{c\alpha}(\gamma) &= \frac{1}{NT_c}\int_0^{NT_c} c(t+\gamma)\alpha(t)dt \\ &= \sum_{m=-\frac{N-3}{2}}^{\frac{N-3}{2}} \left[\tfrac{N-1}{2} - |m|\right] R_{cc}(\gamma+mT_c) & (6) \end{aligned}$$

where $R_{cc}(.)$ is the periodic autocorrelation function of the binary m-sequence, given by

$$R_{cc}(\gamma) = \begin{cases} 1 - \frac{N+1}{NT_c}|\gamma|\,, & |\gamma| \le T_c \\ -\frac{1}{N}\,, & T_c < |\gamma| \le \frac{NT_c}{2} \end{cases}$$

For $\gamma \in \left[-\frac{N-1}{2}T_c, \frac{N+1}{2}T_c\right]$, $R_{c\alpha}(\gamma)$ is given by

$$R_{c\alpha}(\gamma) = \begin{cases} \frac{(N-1)(N+3)}{4N} - \frac{N+1}{NT_c}|\gamma|\,, & |\gamma| \le \frac{(N-1)T_c}{2} \\ -\frac{(N-1)^2}{4N}\,, & \frac{(N-1)T_c}{2} < \gamma \le \frac{(N+1)T_c}{2} \end{cases} \tag{7}$$

$R_{cc}(.)$ and $R_{c\alpha}(.)$ have the same period NT_c. $R_{c\alpha}(\gamma)$ is plotted in Figure 3.

The middle and the lower parts in Figure 2 are the voltage controlled clock (VCC) and voltage controlled oscillator (VCO) loops, respectively. Both loops correlate the received signal $u(t)$ with the difference of the early and late versions of the local code waveform $\alpha(t)$. The difference signal is delayed by $\frac{NT_c}{4}$ seconds in the VCC loop. Both loops have a correlation filter with a frequency response $H(f)$, corresponding to a rectangular shape impulse response of duration MT_c seconds. The correlation filter outputs are sampled at every T_c seconds. The VCC loop uses the carrier phase estimate from the VCO loop and the VCO loop uses the code phase estimate from the VCC loop. Hence, the two loops work jointly. The code and the carrier phase estimates are updated once every T_c seconds with the following recursive equations:

$$\begin{aligned} \hat{\tau}_{m+1} &= \hat{\tau}_m + K_{vcc}\mathrm{sgn}(y_{1,m+1}(\hat{\tau}_m, \tau, \hat{\phi}_m, \phi)) && (8) \\ \hat{\phi}_{m+1} &= \hat{\phi}_m + K_{vco}\mathrm{sgn}(y_{2,m+1}(\hat{\tau}_m, \tau, \hat{\phi}_m, \phi)) && (9) \end{aligned}$$

where $\hat{\tau}_m$ and $\hat{\phi}_m$ are, respectively, the code and the carrier phase estimates formed at $t = mT_c$, and sgn(.) denotes the sign operator. The signals $y_{1,m+1}(\hat{\tau}_m, \tau, \hat{\phi}_m, \phi)$ and $y_{2,m+1}(\hat{\tau}_m, \tau, \hat{\phi}_m, \phi)$ are the real part and the imaginary part operator outputs, respectively, at $t = (m+1)T_c$. K_{vcc} and K_{vco} are the constants that determine the step sizes of the updates in (8) and (9), respectively. All the signals shown in Figure 2 correspond to the time $t = mT_c$. At $t = mT_c$, code phase estimation error $e_{\tau,m}$ and carrier phase estimation error $e_{\phi,m}$ are defined as

$$\begin{aligned} e_{\tau,m} &= \hat{\tau}_m - \tau && (10) \\ e_{\phi,m} &= \hat{\phi}_m - \phi && (11) \end{aligned}$$

The errors (10) and (11) satisfy the recursions (8) and (9), respectively. We can write $y_{1,m+1}(\hat{\tau}_m, \tau, \hat{\phi}_m, \phi)$ and $y_{2,m+1}(\hat{\tau}_m, \tau, \hat{\phi}_m, \phi)$ as

$$\begin{aligned} y_{1,m+1}(\hat{\tau}_m, \tau, \hat{\phi}_m, \phi) &= \frac{1}{M} \sum_{k=m+1-M}^{m} [s_{1,k+1}(\hat{\tau}_k, \tau, e_{\phi,k}) + \eta_{1,k+1}] && (12) \\ y_{2,m+1}(\hat{\tau}_m, \tau, \hat{\phi}_m, \phi) &= \frac{1}{M} \sum_{k=m+1-M}^{m} [s_{2,k+1}(\hat{\tau}_k, \tau, e_{\phi,k}) + \eta_{2,k+1}] && (13) \end{aligned}$$

where $s_{1,k+1}(\hat{\tau}_k, \tau, e_{\phi,k})$, $s_{2,k+1}(\hat{\tau}_k, \tau, e_{\phi,k})$, $\eta_{1,k+1}$, and $\eta_{2,k+1}$ are defined as

$$\begin{aligned} s_{1,k+1}(\hat{\tau}_k, \tau, e_{\phi,k}) &= \cos(e_{\phi,k})\sqrt{2P} \int_{kT_c}^{(k+1)T_c} c(t-\tau)x(t-\hat{\tau}_k)dt \\ s_{2,k+1}(\hat{\tau}_k, \tau, e_{\phi,k}) &= -\sin(e_{\phi,k})\sqrt{2P} \int_{kT_c}^{(k+1)T_c} c(t-\tau)x(t-\hat{\tau}_k+\tfrac{NT_c}{4})dt \\ \eta_{1,k+1} &= \mathrm{Re}\left\{ e^{-j\hat{\phi}_k} \int_{kT_c}^{(k+1)T_c} z(t)x(t-\hat{\tau}_k)dt \right\} \\ \eta_{2,k+1} &= \mathrm{Im}\left\{ e^{-j\hat{\phi}_k} \int_{kT_c}^{(k+1)T_c} z(t)x(t-\hat{\tau}_k+\tfrac{NT_c}{4})dt \right\} && (14) \end{aligned}$$

where $x(t) = \alpha(t - \frac{T_c}{2}) - \alpha(t + \frac{T_c}{2})$.

Over time the signals $y_{1,m}$ and $y_{2,m}$ follow their average values $\overline{y}_1$ and $\overline{y}_2$, respectively. For $M \neq 1$, derivation of these average signals are very difficult because both $y_{1,m}$ and $y_{2,m}$ depend on M previous code and carrier phase estimates. For $M = 1$, the average signals $\overline{y}_1$ and $\overline{y}_2$ are found as

$$\begin{aligned} \overline{y}_1 &= \cos(e_\phi)\sqrt{2P}T_c \left[R_{c\alpha}\left(e_\tau + \tfrac{T_c}{2}\right) - R_{c\alpha}\left(e_\tau - \tfrac{T_c}{2}\right)\right] && (15) \\ \overline{y}_2 &= -\sin(e_\phi)\sqrt{2P}T_c \left[R_{c\alpha}\left(e_\tau - \tfrac{NT_c}{4} + \tfrac{T_c}{2}\right) - R_{c\alpha}\left(e_\tau - \tfrac{NT_c}{4} - \tfrac{T_c}{2}\right)\right] && (16) \end{aligned}$$

and are plotted in Figures 4 and 5, respectively. In these figures, $e_\tau = \hat{\tau} - \tau$ and $e_\phi = \hat{\phi} - \phi$. It is observed from these plots and the recursive equations (8) and (9) that the code and the carrier phase estimation errors converge to either (0,0) or $(\frac{NT_c}{2}, \pi)$ in the (e_τ, e_ϕ) space depending on the initial error coordinate $(e_{\tau,0}, e_{\phi,0})$ at $t = 0$. If $\frac{NT_c}{4} < |e_{\tau,0}| \leq \frac{NT_c}{2}$ and $\frac{\pi}{2} < |e_{\phi,0}| \leq \pi$, then the joint estimates converge to $(\frac{NT_c}{2}, \pi)$. If $|e_{\tau,0}| \leq \frac{NT_c}{4}$ and $|e_{\phi,0}| \leq \frac{\pi}{2}$, then the joint estimates converge to $(0,0)$. For the other initial errors, convergence point depends on whether or not $|e_{\tau,0}|$ crosses $\frac{NT_c}{4}$ before $|e_{\phi,0}|$ crosses $\frac{\pi}{2}$.

The upper part in Figure 2 is the code phase alignment detector. It has a separate PN signal generator and has two noncoherent detectors. The lower detector checks if the incoming code phase is aligned with the local code phase to within $\frac{T_c}{2}$. The upper detector uses a $\frac{NT_c}{2}$ seconds delayed version of the PN signal to check if the incoming and local code phases differ by $\frac{NT_c}{2}$ to within $\frac{T_c}{2}$ seconds. Both detectors correlate the incoming signal with their PN signals for a fixed dwell-time of $n_2 T_c$ seconds. A new value of the code phase estimate is fed to the detectors every $n_2 T_c$ seconds, i.e., at $m = 0, n_2, 2n_2, \ldots$, and is tested for phase alignment. This continues until alignment is declared which will initiate the tracking circuitry. The larger of the two detector outputs is used in the threshold testing. If the threshold is exceeded and if the lower detector output is the larger one, then the joint code and carrier phase tracking circuitry is activated with the present code and carrier phase estimates. However, if the upper detector output is the larger one with a threshold crossing, then the code and the carrier phase estimates are adjusted by $\frac{NT_c}{2}$ and π, respectively, before activating the joint code and carrier phase tracking circuitry.

There is a code phase alignment if the incoming and the local code phases are within $\frac{T_c}{2}$ of each other or if they are separated by more than $\frac{(N-1)T_c}{2}$ and less than $\frac{NT_c}{2}$ seconds. Detection probability P_d is defined as the probability of declaring acquisition when the two code phases are in fact aligned (H_1). Similarly, P_{fa} is defined as the probability of falsely declaring acquisition when the code phases are not aligned (H_0). The hypotheses H_1 and H_0, for each integer l, are defined as

$$\begin{aligned} &H_1 : |\hat{\tau}_{ln_2} - \tau| \leq \tfrac{T_c}{2} \quad \text{or} \quad \tfrac{N-1}{2}T_c \leq |\hat{\tau}_{ln_2} - \tau| \leq \tfrac{N}{2}T_c \qquad \text{(alignment)} \\ &H_0 : T_c \leq |\hat{\tau}_{ln_2} - \tau| \leq \tfrac{N-2}{2}T_c \qquad \text{(no alignment)} \end{aligned}$$

3 Performance Analysis

We evaluate the mean and the variance of the acquisition time of the proposed receiver using the flow graph technique [4]. The flow graph for acquisition process is shown in Figure 6, where the states are defined as

$$\begin{array}{lll} \text{state } (0,0), & |e_{\tau,m}| \leq \frac{T_c}{2} \text{ and } |e_{\phi,m}| \leq \frac{0.5\pi}{K-0.5} & \\ \text{state } (0,j), & |e_{\tau,m}| \leq \frac{T_c}{2} \text{ and } \frac{\pi(j-0.5)}{K-0.5} < |e_{\phi,m}| \leq \frac{\pi(j+0.5)}{K-0.5}\,, \; j = 1,2,\ldots,K-1 & \\ \text{state } (i,0), & (i-0.5)T_c < |e_{\tau,m}| \leq (i+0.5)T_c \text{ and } |e_{\phi,m}| \leq \frac{0.5\pi}{K-0.5} & \\ & \qquad i = 1,2,\ldots,\frac{N-1}{2} & \\ \text{state } (i,j), & (i-0.5)T_c < |e_{\tau,m}| \leq (i+0.5)T_c \text{ and } \frac{\pi(j-0.5)}{K-0.5} < |e_{\phi,m}| \leq \frac{\pi(j+0.5)}{K-0.5} & \\ & \qquad i = 1,2,\ldots,\frac{N-1}{2} \text{ and } j = 1,2,\ldots,K-1 & \\ & & \\ \text{state } S, & \text{start} \quad \text{(start acquisition)} & \\ \text{state } F, & \text{finish} \quad \text{(achieve acquisition)} & \end{array} \tag{17}$$

The system can start at any of the states (i,j), $i = 0, 1, ..., \frac{N-1}{2}$ and $j = 0, 1, ..., K-1$, with the following probabilities.

$$P_{i,j} = \Pr\{S = (i,j)\} = \begin{cases} \frac{1}{N(2K-1)} & i = 0, j = 0 \\ \frac{2}{N(2K-1)} & i = 0, j \neq 0 \ \text{ or } \ i \neq 0, j = 0 \\ \frac{4}{N(2K-1)} & i \neq 0, j \neq 0 \end{cases} \tag{18}$$

The transitions among the states (i,j), $i = 0, 1, ..., \frac{N-1}{2}$ and $j = 0, 1, ..., K-1$, that are not labeled in Figure 6 have a transfer function $H(z)$ where $H(z) = P_{fa}z^{K_p+1} + (1-P_{fa})z$. The functions $H_2(z)$ and $H_3(z)$ are defined as $H_2(z) = P_d z^{K_p+1} + (1-P_d)z$ and $H_3(z) = (1-P_d)zH^{2\nu-1}(z)$. In Figure 6, K is assumed to be an odd integer and z represents the unit delay operator. A unit delay for the proposed receiver is n_2T_c seconds. Assume that the penalty time due to a false alarm is $K_pn_2T_c$ seconds and the peak swing of the code phase estimation error when code phase alignment is not detected, is νT_c. The correlation filter parameter M and the VCC loop step size K_{vcc} determine the value of ν [5]. For a fixed K_{vcc}, ν increases as M increases. Although a small ν, i.e., a small M, is desired for a shorter average acquisition time, the correlation filter with a small M has significant fluctuations at its output due to noise and code chips, giving unsatisfactory operation. Therefore, M and ν should be sufficiently large. For a code period $N = 511$, $\nu \geq 10$ yields satisfactory performance. The penalty time of activating the tracking circuitry with a carrier phase error larger than $\frac{\pi}{2}$ is also modeled as $K_pn_2T_c$ seconds. Some of the intermediate states and the transitions between the states $(\frac{N-1}{2}, j)$, $j = \frac{K+1}{2}, ..., K-1$, are not shown in Figure 6 for simplicity. A detailed description of the flow graph along with the transfer functions $G_{\frac{N-1}{2},\frac{K+1}{2}}(z), ..., G_{\frac{N-1}{2},K-1}(z)$ is given in [5]. In Figure 6, $\frac{K-1}{2}$ is assumed to be an even integer. If $\frac{K-1}{2}$ is odd then there is a transition from $(0, \frac{K-1}{2})$ to $(0,1)$ instead of the transition to $(0,0)$. The following results are valid regardless of $\frac{K-1}{2}$ being even or odd.

We use the generating function technique of [4] in our analysis. The generating function $U(z)$ is

$$U(z) = \sum_{m=0}^{\infty} P_{SF}(m)z^m$$

where, $P_{SF}(m)$ is the probability of going from state S to state F in m steps. $U(z)$ is also the transformed signal at node F when a unit input is applied to node S. Therefore, we can write

$$U(z) = \sum_{j=0}^{K-1} \sum_{i=0}^{\frac{N-1}{2}} P_{i,j}G_{i,j}(z) \tag{19}$$

where $G_{i,j}(z)$ is the transfer function between state (i,j) and F. The functions $G_{i,j}(z)$ for $i = 0, 1, ..., \frac{N-1}{2}$ and $j = 0, 1, ..., K-1$ are given in [5]. The average acquisition time $\overline{T}_{joint,cl}$ (cl denotes closed-loop) for the proposed receiver can be derived from $U(z)$ as

$$\overline{T}_{joint,cl} = \left[\sum_{m=0}^{\infty} mP_{SF}(m)\right] nT_c = \left.\frac{dU(z)}{dz}\right|_{z=1} n_2T_c \tag{20}$$

Substituting (19) into (20) we obtain,

$$\begin{aligned} \overline{T}_{joint,cl} = {} & \frac{n_2T_c}{24N(2K-1)P_d}\Big\{24(2K-1)\left(N + P_d + P_d^2K_p\right) + \\ & (P_{fa}K_p + 1)\,[24N(1-2K) + 48N\nu(2K-1)(1-P_d) + \\ & P_d(3N^2(2K-1) + 12N(K^2+5K-2) - 8K^3 - 12K^2 - 166K + 87)]\Big\} \end{aligned} \tag{21}$$

This average acquisition time is compared with those of the noncoherent single-dwell serial-search receiver in Figure 1, the coherent closed-loop acquisition receiver and the coherent single-dwell serial-search receiver in [6]. The average acquisition time of the noncoherent serial system is obtained in [4] as

$$\overline{T}_{noncoh,serial} = \frac{2 + (2 - P_d)(N-1)(P_{fa}K_p + 1)}{2P_d} n_1 T_c \tag{22}$$

The average acquisition time of the coherent serial system, $\overline{T}_{coh,serial}$, is similar to (22) with $n_1 T_c$ being replaced with the coherent detector dwell-time $n_3 T_c$. The average acquisition time of the coherent closed-loop acquisition receiver was obtained in [6] as

$$\overline{T}_{coh,cl} = \left[(P_{fa}K_p + 1)\frac{N^2 - 1}{4N} + \frac{1 + (1 - P_d)(2\nu - 1)(P_{fa}K_p + 1)}{P_d} \right] n_3 T_c \tag{23}$$

To compare the average acquisition times, we define

$$R_1 = \frac{\overline{T}_{noncoh,serial}}{\overline{T}_{joint,cl}} \quad , \quad R_2 = \frac{\overline{T}_{coh,serial}}{\overline{T}_{joint,cl}} \quad , \quad \text{and} \quad R_3 = \frac{\overline{T}_{coh,cl}}{\overline{T}_{joint,cl}} \tag{24}$$

To obtain $P_d = 0.9$ and $P_{fa} = 0.1$ at SNR=-5 dB, we use $n_1 = 130$, $n_2 = 175$ and $n_3 = 87$ [5]. R_1, R_2 and R_3 are plotted for various values of ν in Figure 7 for $N = 511$, $K_p = 10$, $K = 5, 19$, SNR=-5 dB, $P_d = 0.9$ and $P_{fa} = 0.1$. The value of ν is in the range $10 \leq \nu \leq 50$ with high probability. For this range of ν it is observed that the joint system is slightly slower than the coherent closed-loop acquisition receiver, 2.8-3.4 times faster than the noncoherent serial system, and 1.9-2.3 times faster than the coherent serial system. This is because the carrier phase recovery helps the code phase acquisition in the joint system, whereas the noncoherent system operates independent of the carrier phase, and the search is unidirectional in both the coherent and the noncoherent serial systems.

The number of the carrier phase error states, K, is determined by the VCO loop step size K_{vco}. As K_{vco} gets smaller, K gets larger and the average acquisition time of the joint system gets longer. As K_{vco} gets larger, i.e., K gets smaller and the system results in a larger carrier phase estimation error when the code phase acquisition is achieved. We computed the probability of reaching acquisition (state F) from states (0,0), (0,1), or $(\frac{N-1}{2}, K-1)$. This is the probability of having the carrier phase estimation error magnitude being in the range $[0, \frac{3\pi}{2K-1}]$ or $[\frac{(2K-3)\pi}{2K-1}, \pi]$ when the code phase acquisition is accomplished. This probability is found as 0.9992 for $K = 5$ and 0.9785 for $K = 19$.

The variance of the acquisition time of the proposed receiver can also be derived from $U(z)$:

$$\sigma^2_{joint,cl} = \left[\frac{d^2U(z)}{dz^2} + \frac{dU(z)}{dz} - \left(\frac{dU(z)}{dz} \right)^2 \right]_{z=1} (n_2 T_c)^2 \tag{25}$$

Substituting (19) into (25) yields [5]

$$\begin{aligned} \sigma^2_{joint,cl} \; = \; & \frac{(n_2T_c)^2}{576(-1+2K)^2N^2P_d^2} \{ P_d^2 \big[-((-1+2K)^2(63+8K+4K^2)^2) \\ & + 24(1-2K)(129-198K+4K^2+2K^4)N + 6(-95+212K-104K^2+24K^4)N^2 \\ & \quad +3(-1+2K)^2N^4 \big] \\ & + 48P_d^3K_p(-1+2K) \big[(-1+2K)(63+8K+4K^2) + 12K(1+K)N - 3(-1+2K)N^2 \big] \\ & + 24NP_d^2P_{fa}K_p^2 \big[-(-1+2K)^2(87+8K+4K^2) + 12(-1+2K)(-2+5K+K^2)N \\ & \quad +3(-1+2K)^2N^2 \big] \end{aligned}$$

$$\begin{aligned}
&+2P_d^2P_{fa}K_p\big[-(-1+2K)^2(63+8K+4K^2)(87+8K+4K^2)\\
&\qquad +24(1-2K)(129-186K+16K^2+2K^4)N\\
&\qquad +6(-83+164K-56K^2+24K^4)N^2+3(-1+2K)^2N^4\big]\\
&-144N^2P_dP_{fa}K_p^2(-1+2K)^2(4+P_d^2)+576N^2P_{fa}^2K_p^2(-1+2K)^2\\
&+P_d^2P_{fa}^2K_p^2\big[-(-1+2K)^2(87+8K+4K^2)^2\\
&\qquad +48(1-2K)(120-194K+8K^2-4K^3+K^4)N\\
&+6(-167+548K-440K^2-96K^3+24K^4)N^2-72(-1+2K)^2N^3+3(-1+2K)^2N^4\big]\\
&+48P_d^3K_p^2(-1+2K)\big[12(-1+2K)(N-P_d)+P_{fa}(-87+166K+12K^2+8K^3\\
&\qquad +12(2-3K+K^2)N)\big]\\
&+1152N^2\nu P_{fa}K_p(-1+2K)^2(1-P_d)\left[P_dK_p(1-P_{fa})-2(1+P_{fa}K_p)\right]\\
&+2304N^2\nu^2(-1+2K)^2(1-P_d)(1+P_{fa}K_p)^2\big\}
\end{aligned}\tag{26}$$

The acquisition time variances of the other systems are given in [6]. Let

$$R_4=\sigma^2_{noncoh,serial}/\sigma^2_{joint,cl}\quad,\quad R_5=\sigma^2_{coh,serial}/\sigma^2_{joint,cl}\quad,\quad \text{and}\quad R_6=\sigma^2_{coh,cl}/\sigma^2_{joint,cl}\tag{27}$$

These ratios are plotted for various values of ν in Figure 8 for $N=511$, $K_p=10$, $K=5,19$, SNR=-5 dB, $P_d=0.9$ and $P_{fa}=0.1$. SNR is the chip signal-to-noise ratio defined as SNR=$\frac{2PT_c}{N_o}$. It is observed that the acquisition-time variance of the joint system is slightly larger than the coherent closed-loop acquisition receiver, but it is significantly smaller than the noncoherent serial system. The joint system also has a smaller acquisition-time variance than the coherent serial system. The acquisition time variance of the joint system gets larger as K gets larger, i.e., as K_{vco} gets smaller.

In conclusion, the proposed system performs better than both the noncoherent and the coherent serial acquisition systems, and almost as good as the coherent closed-loop acquisition system. The tradeoff is the added complexity of the proposed system.

References

[1] K. Simon, J. Omura, R. Scholtz, and K. Levitt, *Spread Spectrum Communication*, Volume III, Computer Science Press, 1985.

[2] R. De Gaudenzi and M. Luise, "Decision-directed coherent delay-lock tracking loop for DS spread-spectrum signals," *IEEE Trans. Comm.*, pp. 758-765, May 1991.

[3] R. A. Iltis and A. W. Fuxjaeger, "Spread-spectrum code, doppler, and multipath tracking using Gaussian sum filtering," *Proc. of the 25th Annual Asilomar Conference on Signals, Systems and Computers*, pp. 143-147, Pacific Grove, CA, Nov. 1991.

[4] J. K. Holmes and C. C. Chen, "Acquisition time performance of PN spread-spectrum systems," *IEEE Trans. Comm.*, pp. 778-783, Aug. 1977.

[5] M. Salih, "Closed-Loop Code Phase Synchronization and Carrier Phase Recovery for DS/SS Systems," Ph.D. Dissertation, University of Massachusetts at Amherst, in preparation.

[6] M. Salih and S. Tantaratana, "A closed-loop coherent PN acquisition scheme for DS/SS systems using an auxiliary sequence," *Proc. of the 28th Annual Asilomar Conference on Signals, Systems and Computers*, pp. 99-103, Pacific Grove, CA, Nov. 1994.

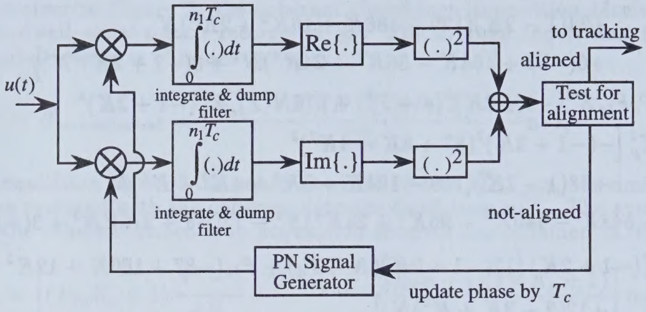

u(t)
$\int_0^{n_1 T_c} (.)dt$
integrate & dump filter
Re{.}
Im{.}
$(.)^2$
to tracking
aligned
not-aligned
Test for alignment
PN Signal Generator
update phase by T_c

Figure 1. Noncoherent, fixed-dwell, serial-search acquisition receiver

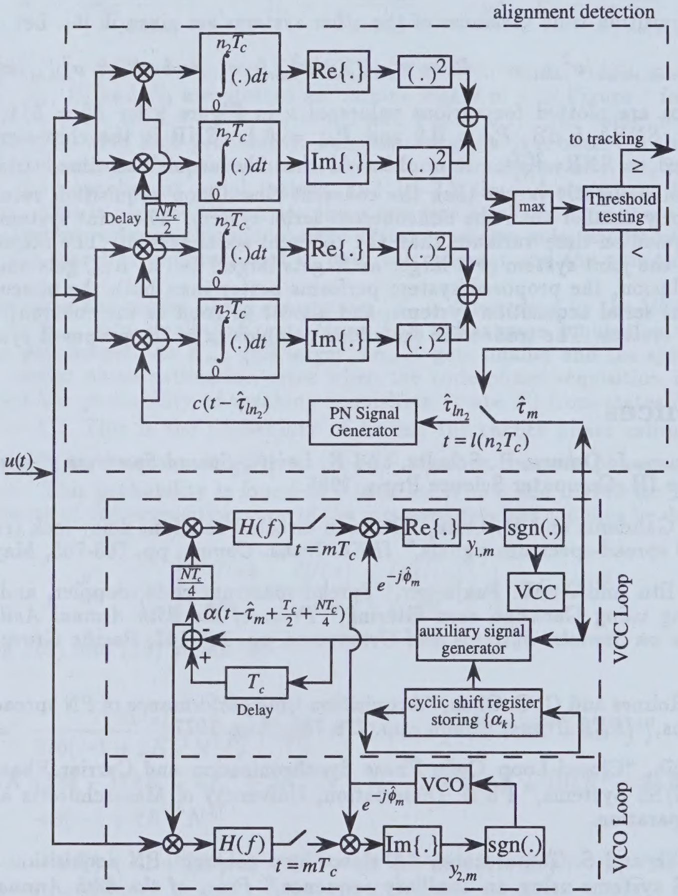

alignment detection
$\int_0^{n_2 T_c} (.)dt$
Re{.}
Im{.}
$(.)^2$
max
Threshold testing
to tracking
≥
<
Delay
$\frac{NT_c}{2}$
$c(t-\hat{\tau}_{ln_2})$
PN Signal Generator
$\hat{\tau}_{ln_2}$
$\hat{\tau}_m$
$t = l(n_2 T_c)$
u(t)
H(f)
$t = mT_c$
$e^{-j\hat{\phi}_m}$
sgn(.)
$y_{1,m}$
VCC
VCC Loop
$\frac{NT_c}{4}$
$\alpha(t-\hat{\tau}_m+\frac{T_c}{2}+\frac{NT_c}{4})$
-
+
auxiliary signal generator
T_c
Delay
cyclic shift register storing $\{\alpha_k\}$
VCO
VCO Loop
$y_{2,m}$

Figure 2. The proposed system

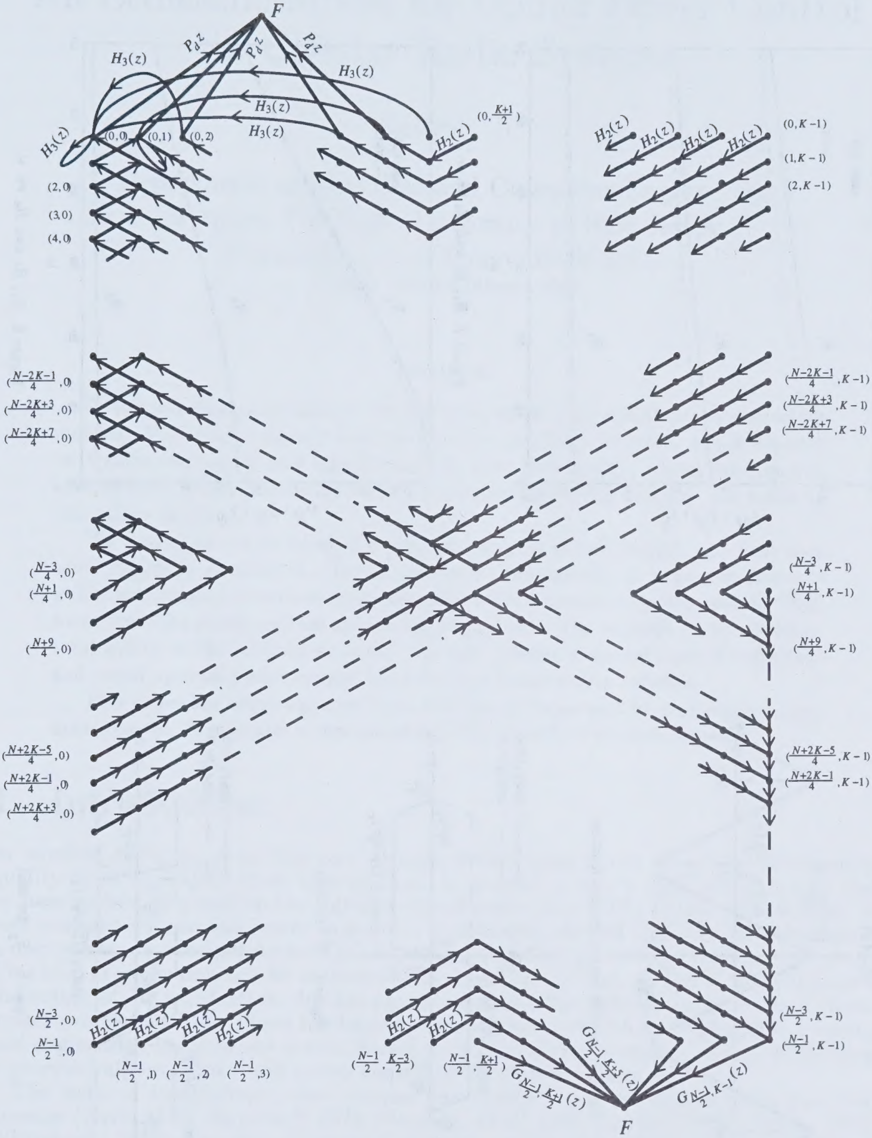

F
$P_d z$
$P_d z$
$P_d z$
$H_3(z)$
$H_3(z)$
$H_3(z)$
$H_3(z)$
$H_3(z)$
(0,0)
(0,1)
(0,2)
(2,0)
(3,0)
(4,0)
$(0,\frac{K+1}{2})$
$H_2(z)$
$H_2(z)$
$H_2(z)$
$H_2(z)$
$H_2(z)$
$(0,K-1)$
$(1,K-1)$
$(2,K-1)$
$(\frac{N-2K-1}{4},0)$
$(\frac{N-2K+3}{4},0)$
$(\frac{N-2K+7}{4},0)$
$(\frac{N-2K-1}{4},K-1)$
$(\frac{N-2K+3}{4},K-1)$
$(\frac{N-2K+7}{4},K-1)$
$(\frac{N-3}{4},0)$
$(\frac{N+1}{4},0)$
$(\frac{N+9}{4},0)$
$(\frac{N-3}{4},K-1)$
$(\frac{N+1}{4},K-1)$
$(\frac{N+9}{4},K-1)$
$(\frac{N+2K-5}{4},0)$
$(\frac{N+2K-1}{4},0)$
$(\frac{N+2K+3}{4},0)$
$(\frac{N+2K-5}{4},K-1)$
$(\frac{N+2K-1}{4},K-1)$
$(\frac{N-3}{2},0)$
$(\frac{N-1}{2},0)$
$H_2(z)$
$H_2(z)$
$H_2(z)$
$H_2(z)$
$(\frac{N-1}{2},1)$
$(\frac{N-1}{2},2)$
$(\frac{N-1}{2},3)$
$H_2(z)$
$H_2(z)$
$(\frac{N-1}{2},\frac{K-3}{2})$
$(\frac{N-1}{2},\frac{K+1}{2})$
$G_{\frac{N-1}{2},\frac{K+1}{2}}(z)$
$G_{\frac{N-1}{2},\frac{K+5}{2}}(z)$
$G_{\frac{N-1}{2},K-1}(z)$
$(\frac{N-3}{2},K-1)$
$(\frac{N-1}{2},K-1)$
F

Figure 6. System Flow Graph

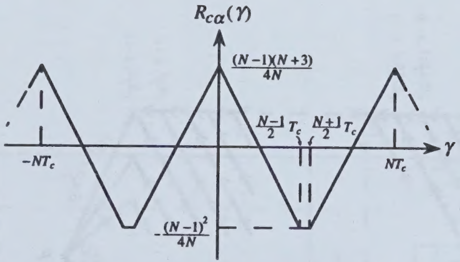

$R_{c\alpha}(\gamma)$
$\frac{(N-1)(N+3)}{4N}$
$\frac{N-1}{2}T_c$
$\frac{N+1}{2}T_c$
γ
$-NT_c$
NT_c
$-\frac{(N-1)^2}{4N}$

Figure 3. Periodic cross-correlation function

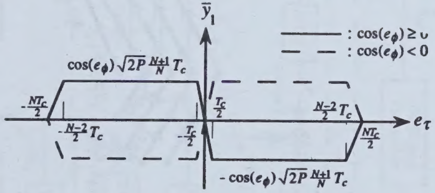

$\overline{y}_1$
$: \cos(e_\phi) \geq 0$
$: \cos(e_\phi) < 0$
$\cos(e_\phi)\sqrt{2P}\,\frac{N+1}{N}T_c$
$-\frac{NT_c}{2}$
$-\frac{N-2}{2}T_c$
$\frac{T_c}{2}$
$-\frac{T_c}{2}$
$\frac{N-2}{2}T_c$
$\frac{NT_c}{2}$
e_τ
$-\cos(e_\phi)\sqrt{2P}\,\frac{N+1}{N}T_c$

Figure 4. $\overline{y}_1$ for M=1

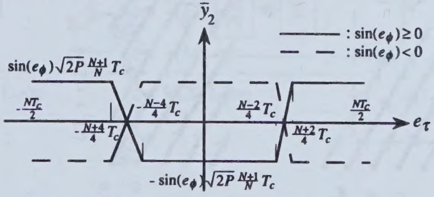

$\overline{y}_2$
$: \sin(e_\phi) \geq 0$
$: \sin(e_\phi) < 0$
$\sin(e_\phi)\sqrt{2P}\,\frac{N+1}{N}T_c$
$-\frac{NT_c}{2}$
$-\frac{N-4}{4}T_c$
$\frac{N-2}{4}T_c$
$\frac{NT_c}{2}$
$-\frac{N+4}{4}T_c$
$\frac{N+2}{4}T_c$
e_τ
$-\sin(e_\phi)\sqrt{2P}\,\frac{N+1}{N}T_c$

Figure 5. $\overline{y}_2$ for M=1

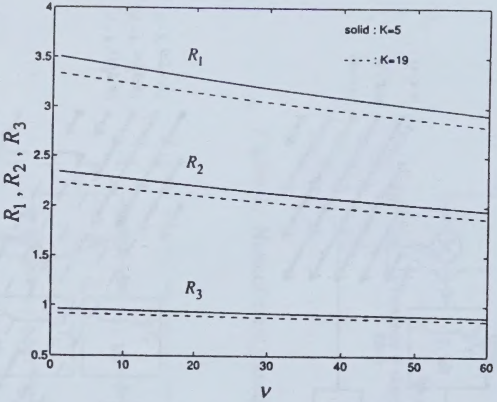


Figure 7. R_1, R_2 and R_3 vs. ν

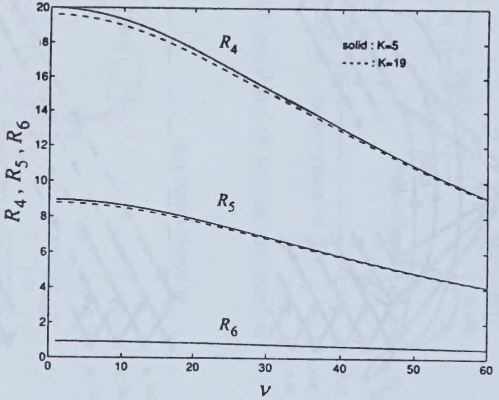


Figure 8. R_4, R_5 and R_6 vs. ν

An Economic Model for Uplink Power Control in Cellular Radio Systems

Hongbin Ji

Department of Electrical and Computer Engineering
Rutgers, The State University of New Jersey
Piscataway, New Jersey 08855-0909
Email: hji@ece.rutgers.edu

Abstract

In this paper, an economic model is originated to study uplink power control for mobiles. We define a mobile user's preference by utility function, which depends on transmitter power and signal-to-interference ratio (SIR). Three fundamental assumptions, which have practical meaning and engineering insights, are made on this utility function.

We formulate the problem of individual optimal power control such that each user's utility is maximized. Under the three assumptions, it is proved that the utility is a component-wise concave function of the transmitter power overall. Then we address the social optimal power control problem. The objective is to maximize total utility of the wireless network. We also present a special case of individual and social optimal power control by defining a linear utility function.

This paper has setup a general economic-based framework for studying resource management. It indicates a new direction for research in wireless network.

1 Introduction

In wireless network, a mobile user adjusts transmitter power to obtain satisfactory quality-of-service (QoS) from base station. In general, a user's QoS, which is provided by base station, depends on the signal-to-interference ratio (SIR) at that base station. A user prefers to use as less power as possible to maintain desired QoS. On the other hand, a user prefers to obtain a better QoS from a base station given fixed transmitter power. This kind of preference can be represented by a utility function. A user's utility indicates the extent of his satisfaction. In this paper, we define the utility as a function of transmitter power and SIR. Three fundamental assumptions, which have practical meaning and engineering insights, are made. Based on this proposed economic model, we develop a general framework to study power control of wireless networks.

The issue of transmitter power control has drawn a lot attention for the past two decades (Aein, 1973; Meyerhoff, 1974; Nagatsu, et al, 1983; Zander, 1992a, 1992b, 1993; Whitehead, 1993; Grandhi, Vijayan, Goodman and Zander, 1993; Yates and Huang, 1995). We study the power control problem from a different point of view. In this paper, an economic model is established to study uplink power control for mobiles. It turns out that this model is so general that it may cover most of the issues concerned with resource management such as power control, base station assignment, call admission control and handoff, etc.

First of all, we make three basic assumptions about utility function. The first assumption is that for given SIR, utility is a monotonically decreasing concave function of transmitter power. The monotonical decrease assumption directly follows from the argument that a user will be less satisfied if he needs more power to maintain a fixed SIR. This is because power is valuable commodity to mobile users. The concavity assumption comes from the fact that changing rate of user's preference will be steeper as larger power is needed to maintain a fixed SIR. This can be further justified by looking at the life time of mobile's battery.

The second assumption assumes that given transmitter power, utility is a monotonically increasing concave function of SIR. As SIR increases, which implies improvement of QoS, users will be happier. When the SIR reaches target, users will be indifferent to any further QoS improvement.

The third assumption assumes that utility is mutually non-increasing of transmitter power and SIR. Under these reasonable assumptions, it is proved that a user's utility is a concave function of his transmitter power overall as all the other users fix their transmitter power.

Then we formulate the problem of individual optimal power control, i.e., each user wants to maximize his utility. An algorithm is proposed to find the equilibrium. We study a special case by defining utility as a linear function of transmitter power and SIR. In this case, there exists an equilibrium.

Finally, we address the social optimal power control problem. The objective is to maximize the total utility of wireless systems. We also present a special case of social optimal power control by defining utility function as a linear function. The result conforms to the work by Zander(1993), Yates and Huang (1995).

This paper has setup a general economic-based framework for pursuing resource management. It indicates a new direction for research in wireless network. The rest of this paper is organized as follows. Section 2 describes the system model first. Then we develop an economic model for resource management of wireless networks. This section has paved the way for us to address individual and social power control optimization problem in Section 3 and Section 4, respectively. Linear utility function is formulated as case study for utility-based power control. The last section concludes the paper and gives perspective of the proposed economic-based general framework.

2 System and Economic Models

Suppose we have N users, M base stations and a common radio channel. The transmitter power of the j-th user is p_j. Assume the j-th user is assigned to the a_j-th base station. For simplicity, we only consider fixed base station assignment in this paper. The case of considering power control and base assignment jointly is presented elsewhere (Ji, 1995).

Let $h_{a_j,j}$ denote the gain from the j-th user to his assigned a_j-th base station. Then, the received signal power at the a_j-th base from the j-th user is given by $h_{a_j,j}p_j$. The interference to the j-th user from the other users at the a_j-th base is given by $\sum_{i\neq j} h_{a_j,i}p_i+\sigma_{a_j}$ where σ_{a_j} denotes the receiver noise power at the a_j-th base, and $h_{a_j,i}$ denotes the gain from the i-th user to the a_j-th base station.

Define $N \times 1$ dimensional power vector, P, and $(N-1)\times 1$ dimensional vector, $\bar{P}_j$:

$$P \triangleq [p_1, p_2, \ldots, p_N]^T, \quad \bar{P}_j \triangleq [p_1, p_2, \ldots, p_{j-1}, p_{j+1}, \ldots, p_N]^T$$

Note that deleting the j-th element of P yields $\bar{P}_j$.

Then we have the SIR of the j-th user at his fixed assigned a_j-th base station, $\gamma_{a_j,j}$:

$$\gamma_{a_j,j} = \frac{h_{a_j,j}p_j}{\sum_{l\neq j} h_{a_j,l}p_l + \sigma_{a_j}} \tag{1}$$

Define the *SIR factor*, denoted by $\mu_{a_j,j}(\bar{P}_j)$, of the j-th user at the a_j-th base station as:

$$\mu_{a_j,j}(\bar{P}_j) \triangleq \frac{h_{a_j,j}}{\sum_{l\neq j} h_{a_j,l}p_l + \sigma_{a_j}} \tag{2}$$

Most of the time, people use P as an argument for a user's SIR factor $\mu_{a_j,j}$. We choose $\bar{P}_j$ instead to indicate that $\mu_{a_j,j}$ is independent of his own transmitter power. This is more precise and clear in notation. In particular, when we study individual or social optimization problems, the partial derivatives of SIR factor will appear a lot of times in the formulae. This will avoid confusion. The SIR of the j-th user at his assigned a_j-th base station is a linear function of his transmitter power if all the other users fix their transmitter power, i.e.

$$\gamma_{a_j,j} = \mu_{a_j,j}(\bar{P}_j)p_j \tag{3}$$

Noticing that the j-th user's SIR factor $\mu_{a_j,j}(\bar{P}_j)$ is independent of the power p_j transmitted by himself, we obtain the derivative of the j-th user's SIR factor with respect to the transmitter power of the i-th user $(i \neq j)$:

$$\frac{\partial \mu_{a_j,j}}{\partial p_i} = -\frac{h_{a_j,j}h_{a_j,i}}{(\sum_{l\neq j} h_{a_j,l}p_l + \sigma_{a_j})^2} = -\frac{h_{a_j,i}}{h_{a_j,j}}\mu^2_{a_j,j}(\bar{P}_j) < 0 \tag{4}$$

Above equation indicates that the SIR factor is a decreasing function of the other users' transmitted power. Moreover, it is a component-wise convex function, i.e., it is convex of the i-th $(i \neq j)$ user's transmitter power p_i when the other users fix their transmitter power. It can be justified by the following inequality:

$$\frac{\partial^2 \mu_{a_j,j}}{\partial p_i^2} = 2\left(\frac{h_{a_j,i}}{h_{a_j,j}}\right)^2 \mu^3_{a_j,j}(\bar{P}_j) > 0 \;\; (i \neq j) \tag{5}$$

We also obtain:

$$\frac{\partial^2 \mu_{a_j,j}}{\partial p_l \partial p_i} = 2\frac{h_{a_j,i}h_{a_j,l}}{h^2_{a_j,j}}\mu^3_{a_j,j}(\bar{P}_j) > 0 \;\; (l \neq i, l \neq j) \tag{6}$$

In general, the SIR factor $\mu_{a_j,j}(\bar{P}_j)$ is not a convex function of vector $\bar{P}_j$. A sufficient condition for $\mu_{a_j,j}$ to be convex of $\bar{P}_j$ is:

$$\sum_{l\neq i, l\neq j} h_{a_j,l} < h_{a_j,i} \tag{7}$$

Obviously, above condition can not be met most of the time. This condition is obtained by checking the positive definiteness of the Hessian matrix $\frac{\partial^2 \mu_{a_j,j}}{\partial \bar{P}_j^2}$. However, the SIR factor $\mu_{a_j,j}$ is always component-wise convex of p_i for $i \neq j$ as shown in equation (5).

Typically, a user wants to maintain his QoS by transmitting a feasible amount of power. As power gain changes when a user moves, power should be adjusted accordingly to compensate the attenuation in the connection link. Power should also be appropriately controlled such that one user will not cause extreme QoS degradation to the other users.

Before a user achieves his target SIR, transmitting more power is attractive to him. After the user has obtained the target SIR, transmitting more power becomes unfavorable. The reason is that users are indifferent to any further QoS improvement after target SIR is achieved, and that power is valuable commodity. This kind of user's preference can be represented by utility function, which reflects users' satisfaction who transmit some amount of power and obtain certain level of quality of service from the network. A user will be more satisfied if he transmits a smaller amount of power to achieve a desired quality of service. On the other hand, he will be happier if he is provided with a better quality of service for a fixed transmitter power.

Definition 2.1 *A user's utility function, denoted by $u_j(p_j, \gamma_{a_j,j})$, depends on his transmitter power p_j and the SIR $\gamma_{a_j,j}$, for $j = 1, 2, ..., N$.*

If the utility function was defined to be dependent only on the transmitter power, it would not reflect users' preference on quality of service. On the other hand, if the utility function was defined to depend only on users' quality of service obtained from the base station, we could not distinguish users' preferences when one user transmits smaller amount of power to obtain the same quality of service as another user who has used more transmitter power. Therefore, it is not sufficient to reflect a user's preference if we define the utility just dependent only on either one of the two parameters. This will further be justified when we develop pricing schemes for wireless networks in (Ji and Huang, 1995a).

We make the following reasonable assumptions about the utility function. These assumptions will imply the utility function is a component-wise concave function of the transmitter power.

Assumption 2.1 *For the j-th user, given SIR $\gamma_{a_j,j}$, the utility function is a monotonically decreasing concave function of his transmitter power for $j = 1, 2, ..., N$.*

Assumption 2.2 *For the j-th user, given his transmitter power p_j, the utility function is a monotonically increasing concave function of the SIR γ_{a_j} for $j = 1, 2, ..., N$.*

Assumption 2.3 *The utility function $u_j(p_j, \gamma_{a_j,j})$ is mixed non-increasing of p_j and $\gamma_{a_j,j}$, i.e. $\frac{\partial^2 u_j}{\partial p_j \partial \gamma_{a_j,j}} \leq 0$, $\forall j$.*

The first assumption says that a user's satisfaction will decrease if he needs to transmit more power to maintain a fixed SIR or keep the quality of service. This is because the power is valuable commodity to users. Moreover, the changing rate of a user's preference will be steeper as more and more transmitter power is needed in order to maintain a fixed SIR. It is said in the second assumption that a user will be more satisfied if he uses a fixed power to get a better quality of service. The changing rate of his preference will decrease as the quality of service keeps on improving. This is due to the fact that users will be indifferent to any further QoS improvement once it reaches target. In general, there is no mixed terms of p_j and $\gamma_{a_j,j}$ in a utility function. Thus, the third assumption can easily be met.

Theorem 2.1 *Under above assumptions, the defined utility function $u_j(p_j, \gamma_{a_j,j})$ is a component-wise concave function of transmitter power p_j overall for $j = 1, 2, ..., N$.*

Proof: Above assumptions give:

$$\frac{\partial u_j}{\partial p_j} < 0, \ \frac{\partial u_j}{\partial \gamma_{a_j,j}} > 0$$

$$\frac{\partial^2 u_j}{\partial p_j^2} < 0, \ \frac{\partial^2 u_j}{\partial \gamma_{a_j,j}^2} < 0; \ \frac{\partial^2 u_j}{\partial p_j \partial \gamma_{a_j,j}} \leq 0 \tag{8}$$

We obtain the first-order partial derivative of the utility function with respect to the transmitter power:

$$\left(\frac{\partial u_j}{\partial p_j}\right) = \frac{\partial u_j(p_j, \gamma_{a_j,j})}{\partial p_j} + \frac{\partial u_j(p_j, \gamma_{a_j,j})}{\partial \gamma_{a_j,j}} \mu_{a_j,j}(\bar{P}_j) \tag{9}$$

Then the second-order partial derivative of the utility function with respect to the transmitter power is given by

$$\left(\frac{\partial^2 u_j}{\partial p_j^2}\right) = \frac{\partial^2 u_j(p_j, \gamma_{a_j,j})}{\partial p_j^2} + 2\frac{\partial^2 u_j}{\partial p_j \partial \gamma_{a_j,j}} \mu_{a_j,j} + \mu_{a_j,j}^2 \frac{\partial^2 u_j}{\partial \gamma_{a_j,j}^2} < 0$$

Above inequality indicates the utility $u_j(p_j, \gamma_{a_j,j})$ is a concave function of the transmit power p_j overall. □

The partial derivatives in the right hand side of equation (9) are shadow price, which represents users' marginal utility of transmitter power and quality of service, respectively. Let us define them by:

$$\zeta_j(P) \triangleq \frac{\partial u_j}{\partial p_j}, \quad \xi_j(P) \triangleq \frac{\partial u_j}{\partial \gamma_{a_j,j}} \tag{10}$$

It should be noted that ζ_j, ξ_j and their derivatives are measurable parameters.

Equation (8) indicates that $\zeta_j(P)$ is negative and a decreasing function of the transmitter power while $\xi_j(P)$ is positive and a decreasing function of the transmitter power, i.e.

$$\frac{\partial(-\zeta_j(P))}{\partial p_j} > 0; \quad \frac{\partial \xi_j(P)}{\partial p_j} < 0 \tag{11}$$

The meaning of the shadow price is clarified when we develop the pricing scheme (Ji and Huang, 1995a). This is out of the scope of this paper.

This section has paved the way for us to develop power control schemes in next two sections.

3 Individual Optimal Power Control

In this section, we study power control from each individual point of view. Each user will choose transmitter power such that a satisfactory QoS is obtained. This problem is called individual power control. It may also be called distributed power control since each user may adjust transmitter power in a distributed way.

Foschini and Miljanic (1993) showed a simple distributed power control that converges at a geometric rate. Mitra (1993) proved geometric convergence for an asynchronous version of Foschini's algorithm. In this paper, we address distributed power control based on the economic model which was developed in the last section.

Each mobile user wants to regulate his transmitter power such that his own utility is maximized. Due to the interference among the users, they compete with each other. Suppose one user raises his transmitter power to improve QoS. The interference to the other users is also increasing. In order to maintain their QoS, the other users must raise their power, which will increase the interference even more. Thus, the QoS improvement is reduced. In this case, we may trap into the situation that we keep on escalating power without improving QoS significantly as expected.

In Section 4, we shall formulate optimization problem from the system viewpoint, i.e. base stations regulate each user's transmitter power such that total utility is maximized.

Now let us formulate the individual optimal power control problem.

3.1 Problem Statement

Each mobile user wants to regulate his power such that his own utility is maximized.

For the j-th user, he wants to find out what is optimal power transmitted by him such that he will be most satisfied. This is a typical individual optimization problem, which is formulated as follows:

$$\max_{p_j} \; u_j(p_j, \gamma_{a_j,j}) \tag{12}$$

for $j = 1, 2, ..., N$. Theorem (2.1) says that the utility function is a concave function of his transmitted power under Assumptions 2.1-2.3. Then when all the other users fix their transmitter power, there is a unique solution. By rewriting equation (9) in terms of the shadow prices defined in equation (10), we have the first-order derivative:

$$\left(\frac{\partial u_j}{\partial p_j}\right) = \zeta_j(P) + \xi_j(P)\mu_{a_j,j}(\bar{P}_j) \tag{13}$$

Then the individual equilibrium should satisfy:

$$\zeta_j(P^*) + \xi_j(P^*)\mu_{a_j,j}(\bar{P}_j^*) = 0 \tag{14}$$

for $j = 1, 2, ..., N$, where P^* is called non-cooperative equilibrium or Nash equilibrium (Nash, 1950). This is a typical N-person game problem (von Neumann and Morgenstern, 1944; Nash, 1950). Since $u_j(p_j, \gamma_{a_j,j})$ is continuous and concave of p_j, there exists a non-cooperative equilibrium (Rosen, 1965). The uniqueness of equilibrium needs further study. We propose an algorithm in the next subsection to obtain the equilibrium.

3.2 Algorithms

By rewriting equation (14), we have:

$$\xi_j(P^*)\mu_{a_j,j}(\bar{P}_j^*) = -\zeta_j(P^*) \tag{15}$$

for $j = 1, 2, ..., N$. It should be noted that the left hand side (LHS) of above equation is a monotonically decreasing function of the transmitter power while the right hand side (RHS) is a monotonically increasing function of the transmitter power. By adjusting transmitter power, an equilibrium will be stricken such that each user's marginal utility of power will balance his marginal utility of SIR times his SIR factor. In other words, at the equilibrium each user's marginal utility of power will equal to his marginal utility of QoS times a factor, which is independent of his power.

On the one hand, suppose RHS of above equation is bigger than LHS for a user. It means his marginal utility of power is bigger than his marginal utility of QoS times an independent factor. In this case, it will be more advantageous for the user to increase his power. On the other hand, if RHS is less than LHS, the user should decrease his power since his current QoS is more than he needs. As developed in (Ji and Huang, 1995a), users will be charged for that over-engineered QoS.

Algorithm 3.1 *Searching Individual Equilibrium:*
Step 1. *Set the initial transmitter power vector P^0, initial iteration step size, $\Delta_j^0 > 0$, for $j = 1, 2, ..., N$, iteration number $t = 0$.*
Step 2. *For $j = 1, 2, ..., N$, iterate:*

$$p_j^{t+1} = p_j^t + \Delta_j^t[\mu_{a_j,j}(\bar{P}_j^t)\zeta_j(P^t) + \xi_j(P^t)] \tag{16}$$

If $[\mu_{a_j,j}(\bar{P}_j^{t+1})\zeta_j(P^{t+1}) + \xi_j(P^{t+1})]$ has the same sign as $[\mu_{a_j,j}(\bar{P}_j^t)\zeta_j(P^t) + \xi_j(P^t)]$, double the step size, i.e. $\Delta_j^{t+1} = 2\Delta_j^t$. Otherwise, decrease the step size by half, i.e. $\Delta_j^{t+1} = \Delta_j^t/2$.
Step 3. *If $\mid P^{t+1} - P^t \mid > \epsilon$, let $t = t + 1$ and go to step 2. Otherwise, stop.*

In above algorithm, ϵ is a $N \times 1$ vector which represents convergence requirement. A more efficient algorithm exists when shadow prices are convex or concave (Ji, Hui and Karasan, 1995).

Note that above algorithm is quite simple and it can be easily implemented in practical wireless systems.

3.3 Special Case: Linear Utility Function

As an illustration, we consider a special case where the utility function is defined as a linear function:

$$u_j(p_j, \gamma_{a_j,j}) \triangleq (p_j^{max} - p_j) + \lambda_j(\gamma_{a_j,j} - \gamma_j^{min}) \tag{17}$$

for $j = 1, 2, ..., N$, where p_j^{max} denotes the maximum power of user j, γ_j^{min} represents the bottom-line of SIR for an acceptable QoS, and λ_j expresses user's relative preference of requiring better QoS or saving his power.

Taking the derivative of the linear utility u_j with respect to p_j, we obtain:

$$-1 + \lambda_j \frac{h_{a_j,j}}{\sum_{l \neq j} h_{a_j,l} p_l + \sigma_{a_j}} = 0 \tag{18}$$

It is interesting to note that the variable p_j is not in above condition. In other words, the transmitter power of all the other users gives the condition for the existence of the optimal transmitted power of the j-th user.

Rewriting above equation, we know that the individual equilibrium satisfies :

$$\sum_{l\neq j}^{N} h_{a_j,l}p_l^* + \sigma_{a_j} = \lambda_j h_{a_j,j} \tag{19}$$

for $j = 1, 2, ..., N$. Let us put above equation into the matrix form:

$$HP^* = B \tag{20}$$

where H, called gain matrix, and B are given by:

$$H = \begin{bmatrix} 0 & \frac{h_{a_1,2}}{h_{a_1,1}} & \cdots & \frac{h_{a_1,N}}{h_{a_1,1}} \\ \frac{h_{a_2,1}}{h_{a_2,2}} & 0 & \cdots & \frac{h_{a_2,N}}{h_{a_2,2}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{h_{a_N,1}}{h_{a_N,N}} & \frac{h_{a_N,2}}{h_{a_N,N}} & \cdots & 0 \end{bmatrix}, \quad B = \begin{bmatrix} \lambda_1 - \frac{\sigma_{a_1}}{h_{a_1,1}} \\ \lambda_2 - \frac{\sigma_{a_2}}{h_{a_2,2}} \\ \vdots \\ \lambda_N - \frac{\sigma_{a_N}}{h_{a_N,N}} \end{bmatrix} \tag{21}$$

Note that H is a non-negative matrix which has some interesting property. It has been explored by Zander (1992) and Grandhi, Vijayan and Goodman (1992), etc.

We study the individual power control under the following constraints elsewhere (Ji and Huang, 1995b):

$$\gamma_{a_j,j} \geq \gamma_j^{min} \tag{22}$$

$$0 \leq p_j \leq p_j^{max} \tag{23}$$

Above constraints specify the feasible set of optimal power control.

4 Social Optimal Power Control

In this section, let us formulate the power control problem from an system viewpoint. The objective is to maximize the total utility instead of aiming at each user's utility.

4.1 Problem Statement

Given each user's preference, we know that the total utility of all the active users, denoted by $J(P)$, is given by the sum:

$$J(P) \triangleq \sum_{j=1}^{N} u_j(p_j, \gamma_{a_j,j}) \tag{24}$$

We formulate the social optimization problem as:

$$\max_{P} \sum_{j=1}^{N} u_j(p_j, \gamma_{a_j,j}) \tag{25}$$

By differentiating equation (25) with respect to p_i, we obtain

$$\frac{\partial J(P)}{\partial p_i} = \frac{\partial u_i}{\partial p_i} + \frac{\partial u_i}{\partial \gamma_{a_i,i}} \frac{h_{a_i,i}}{\sum_{l\neq i} h_{a_i,l}p_l + \sigma_{a_j}} - \sum_{j\neq i} \frac{\partial u_j}{\partial \gamma_{a_j,j}} \frac{h_{a_j,i}h_{a_j,j}p_j}{\left(\sum_{l\neq j} h_{a_j,l}p_l + \sigma_{a_j}\right)^2} \tag{26}$$

for $i = 1, 2, ..., N$. Using equation (2), we simplify above representation into

$$\frac{\partial J(P)}{\partial p_i} = \frac{\partial u_i}{\partial p_i} + \frac{\partial u_i}{\partial \gamma_{a_i,i}} \mu_{a_i,i} - \sum_{j \neq i} \frac{\partial u_j}{\partial \gamma_{a_j,j}} \frac{h_{a_j,i}}{h_{a_j,j}} \mu^2_{a_j,j} p_j \tag{27}$$

Then we have:

$$\left.\frac{\partial J(P)}{\partial p_i}\right|_{P^*} = \zeta_i(P^*) + \xi_i(P^*)\mu_{a_i,i}(\bar{P_i^*}) - \sum_{j \neq i} \xi_j(P^*) \frac{h_{a_j,i}}{h_{a_j,j}} \mu^2_{a_j,j}(\bar{P_j^*}) p_j = 0 \tag{28}$$

for $i = 1, 2, ..., N$, where P^* is social optimum or Walrasian equilibrium in what has come to be known as the Arrow-Debreu economy (Arrow, 1953; Debreu, 1959). The existence and uniqueness of Walrasian equilibrium was studied in (Aubin, 1979).

Equation (28) implies that the i-th user should use the transmitter power until the marginal benefit from the transmitted power equals the marginal cost that he imposes on the other users and himself.

The first term on the right hand side of equation (27) is negative, the second term is positive and the third term is negative. So, it is not clear whether $J(P)$ is a concave or component-wise concave of the transmitter power. The following theorem gives sufficient condition such that $J(P)$ is a component-wise concave function.

Theorem 4.1 *The total utility $J(P)$ is a component-wise concave function of the transmitter power p_i, for $i = 1, 2, ..., N$ under the condition:*

$$\frac{\partial^2 u_j}{\partial \gamma^2_{a_j,j}} \gamma_{a_j,j} + 2\frac{\partial u_j}{\partial \gamma_{a_j,j}} < 0, \forall j \tag{29}$$

Proof: We obtain the second-order partial derivative of $J(P)$ with respect to p_i:

$$\begin{aligned} \frac{\partial^2 J(P)}{\partial p_i^2} = {} & \frac{\partial^2 u_i}{\partial p_i^2} + \mu_{a_i,i}(\bar{P}_i) \frac{\partial^2 u_i}{\partial p_i \partial \gamma_{a_i,i}} + \mu^2_{a_i,i}(\bar{P}_i) \frac{\partial^2 u_i}{\partial \gamma^2_{a_i,i}} \\ & - \sum_{j \neq i} \frac{\partial^2 u_j}{\partial \gamma^2_{a_j,j}} \left(-\frac{h_{a_j,i}}{h_{a_j,j}} \mu^2_{a_j,j} p_j\right) \frac{h_{a_j,i}}{h_{kj}} \mu^2_{a_j,j} p_j \\ & - \sum_{j \neq i} \frac{\partial u_j}{\partial \gamma_{a_j,j}} \frac{h_{a_j,i}}{h_{a_j,j}} p_j \, 2\mu_{a_j,j} \left(-\frac{h_{a_j,i}}{h_{a_j,j}} \mu^2_{a_j,j}\right) \end{aligned} \tag{30}$$

Collecting the common factors in the last two terms yields:

$$\begin{aligned} \frac{\partial^2 J(P)}{\partial p_i^2} = {} & \frac{\partial^2 u_i}{\partial p_i^2} + \mu_{a_i,i}(\bar{P}_i) \frac{\partial^2 u_i}{\partial p_i \partial \gamma_{a_i,i}} + \mu^2_{a_i,i}(\bar{P}_i) \frac{\partial^2 u_i}{\partial \gamma^2_{a_i,i}} \\ & + \sum_{j \neq i} \left(\frac{h_{a_j,i}}{h_{a_j,j}}\right)^2 p_j \mu^3_{a_j,j} \left(\frac{\partial^2 u_j}{\partial \gamma^2_{a_j,j}} \gamma_{a_j,j} + 2\frac{\partial u_j}{\partial \gamma_{a_j,j}}\right) \end{aligned} \tag{31}$$

Combination of the condition (29) and inequalities (8) yields $\frac{\partial^2 J(P)}{\partial p_i^2} < 0$, which means the total utility is a component-wise concave function of the transmitted power $p_i, \forall i$. □

Algorithm 3.1 can be applied here to solve this social optimal control problem.

4.2 Special Case: Linear Utility Function

Each user's utility function is a linear function of his transmitter power and SIR as defined in equation (17). Then the total utility is given by:

$$J(P) = \sum_{j=1}^{N} [(p_j^{max} - p_j) + \lambda_j(\gamma_{a_j,j} - \gamma_j^{min})] \tag{32}$$

Taking partial derivative of J with respect to p_i, we obtain:

$$\frac{\partial J}{\partial p_i} = -1 + \mu_{a_i,i}\lambda_i - \sum_{j \neq i} \lambda_j \frac{h_{a_j,i}}{h_{a_j,j}} \mu^2_{a_j,j} p_j = 0 \tag{33}$$

Rearranging above equation, we have:

$$\sum_{j=1}^{N} \lambda_j \frac{h_{a_j,i}}{h_{a_j,j}} \mu^2_{a_j,j} p_j = 1 - \mu_{a_i,i}\lambda_{a_i,i} + \lambda_i \mu^2_{a_i,i} p_i \tag{34}$$

Let us look at the monotonicity of the LHS by taking its derivatives:

$$\frac{\partial LHS}{\partial p_i} = \lambda_i \mu^2_{a_i,i} + \sum_{j \neq i} \lambda_j \frac{h_{a_j,i}}{h_{a_j,j}} 2\mu_{a_j,j} p_j \left(-\frac{h_{a_j,i}}{h_{a_j,j}} \mu^2_{a_j,j} \right) \tag{35}$$

$$= \lambda_i \mu^2_{a_i,i} - 2 \sum_{j \neq i} \lambda_j \left(\frac{h_{a_j,i}}{h_{a_j,j}} \right)^2 \mu^3_{a_j,j} \tag{36}$$

Therefore, if

$$\lambda_i \mu^2_{a_i,i} < 2 \sum_{j \neq i} \lambda_j \left(\frac{h_{a_j,i}}{h_{a_j,j}} \right)^2 \mu^3_{a_j,j} \tag{37}$$

the LHS of equation (34) is a decreasing function of the transmitter power. Note that the RHS of above equation is an increasing function of the power. They strike a balance at the equilibrium. Above condition is slightly different from that condition in equation (29). This is due to the fact that the linear utility function does not satisfy those three basic assumptions. Most of the time, condition (37) can be met since the left hand side just contains one term while the right hand side is a sum of $N-1$ positive terms. Algorithm 3.1 is applicable to this case.

5 Conclusion and Perspective

This paper has developed an economic model to study uplink power control for wireless network. Both individual and social optimal power control are discussed.

The proposed model can be explored further to study resource management of wireless networks such as call admission control, handoff and base station assignment (Ji, 1995). We are also developing an efficient pricing scheme which will discourage usage of network resources when congestion appears (Ji and Huang, 1995a).

Acknowledgement: The author would like to thank Ching-Yao Huang, Roy Yates, and Jack Holtzman of Rutgers WINLAB for stimulating discussion.

References

[1] Aein, J. M. (1973), Power Balancing in System Employing Frequency Reuse, *COMSAT Technical Review*, vol. 3, no. 2, Fall.

[2] Arrow, K. J. (1965), *Aspects of the theory of Risk-Bearing*, Yrjo Jahnsson Saatio, Helsinki.

[3] Aubin, J. P. (1979), *Mathematical Methods of Game and Economic Theory*, North-Holland Publishing Company, Amsterdam.

[4] Debreu, G., (1959), *Theory of Value: An Axiomatic analysis of General Equilibrium*, Yale University Press, New Haven, NH.

[5] Foschini, G. J. and Z. Miljanic (1993), A Simple Distributed Autonomous Power Control Algorithm and its Convergence, *IEEE Transactions on Vehicular Technology*, vol. 42, no. 4, November.

[6] Grandhi, S. and R. Vijayan and D. J. Goodman, and J. Zander (1993), Centralized Power Control for Cellular Radio Systems, *IEEE Transactions on Vehicular Technology*, vol. 42, no. 4, November.

[7] Ji, H. (1995), Economic model for resource management of wireless networks, to be submitted to *IEEE Transactions on Vehicular Technology.*

[8] Ji, H. and Huang, C. Y. (1995a), Pricing and power control in cellular radio systems, submitted to *IEEE VTC'96*, Atlanta, GA, April 28–May 1, 1996.

[9] Ji, H. and Huang, C. Y. (1995b), Non-cooperative uplink power control in cellular radio systems, submitted to *Wireless Networks*, special issue on transmitter power control in wireless networks.

[10] Ji, H., J. Y. Hui and E. Karasan (1995), GoS-based pricing and resource allocation for multimedia broadband networks, submitted to *INFOCOM'96,* July.

[11] Meyerhoff, H. J. (1974), Method for Computing the Optimum Power Balancing in Multibeam Satellite, *COMSAT Technical Review*, vol. 4, no. 1, Spring.

[12] Mitra, D. (1993), An Asynchronous Distributed Algorithm for Power Control in Cellular Radio Systems, *Fourth WINLAB Workshop on Third Generation Wireless Information Networks.*

[13] Nagatsu, T., T. Tsuruhara, and M. Sakamoto (1983), Transmitter power control for cellular land mobile radio, *IEEE Global Telecommunications Conference, GLOBECOM-83.*

[14] Nash, J. (1950), Equilibrium points in n-person games, *Proc. Nat. Acad. Sci. U.S.A.*, no. 36.

[15] Rosen, J. B. (1965), Existence and uniqueness of equilibrium points for concave N-person games, *Econometrica*, vol. 33, no. 3, July.

[16] von Neumann, J. and O. Morgenstern (1945), *Theory of Games and Economic Behavior*, Princeton University Press, Princeton, NJ.

[17] Whitehead, J. F (1993), Single-Level-Based Dynamic Power Control for Co-channel Interference Management, *Proc. of IEEE Vehicular Technology Conference*, pp 499-502.

[18] Yates, R. and C. Y. Huang (1995), Integrated Power Control and Base Station Assignment, *IEEE Transactions on Vehicular Technology*, to appear.

[19] Yates, R. (1995), A framework for uplink power control in cellular radio systems, *IEEE Journal on Selected Areas in Communications*, to appear.

[20] Zander, J. (1992a), Performance of Optimum Transmitter Power Control in Cellular Radio Systems, *IEEE Transactions on Vehicular Technology*, vol. 41, February.

[21] Zander, J. (1992b), Distributed Cochannel Control in Cellular Radio Systems, *IEEE Transactions on Vehicular Technology*, vol. 41, no. 3, August.

[22] Zander, J. (1993), Transmitter Power Control for Co-channel Interference Management in Cellular Radio Systems, *Fourth WINLAB Workshop on Third Generation Wireless Information Networks.*

INDEX OF AUTHORS

Aazhang, B. 628
Acar, I. 1073
Al-Dhahir, N. 598
Al-Duwaish, H. 340
Al-Semari, S.A. 638
Alexander, P.D. 29
Alpay, M.E. 411
Altman, E. 964, 974
Amato, F. 59, 89
Antsaklis, P.J. 69, 176
Azizõglu, M. 265

Bakkaloǧlu, B. 440
Banerjee, A. 275, 578
Baral, C. 668
Barbeau, M. 96
Barron, A. 1028
Berg, G.A. 483
Berger, T. 873
Blake, I.F. 648, 710
Bonnecaze, A. 908
Brooks, C.A. 106

Calderbank, A.R. 475
Caruso, A. 89
Chandra, A. 552, 588
Chandrasekar, V. 340
Chang, B.C. 49
Chen, C.-H. 116
Chen, T.-T. 225
Chen, S.L. 501
Cheng, J.-F. 366
Chiu, Y.-J. 873
Chong, E.K.P. 126
Chiasson, G.M. 796
Christensen, G.E. 275
Cioffi, J.M. 598
Conway, A.E. 724
Corless, M. 59
Costello, D.J. 357

Dahleh, M.A. 186
de Oliveira Brandão, L. 510
Doerschuk, P.C. 285
Doherty, J.F. 295
Dong, G. 1020
Duel-Hallen, A. 01
Dumer, I. 688
Duursma, I.M. 908

El-Ezabi, A. 01
Elia, N. 186

Fedortsov, S.P. 39
Feng, G.-L. 483
Fialho, I.J. 845
Fitz, M.P. 235, 245, 618
Forney, G.D. 598
Francis, B.A. 401
Fuja, T.E. 638
Fujiwara, T. 700

Gansman, J.A. 235
Garcia, M.P. 510
Garg, V.K. 166
Garofalo, F. 89
Gburzyński, P. 303
Gelfand, S.B. 245, 618
Georgiou, T.T. 845
Ghazi-Moghadam, V. 216
Glielmo, L. 89
Goeckel, D.L. 815
Goldsmith, A. 915
Grant, A.J. 21
Grigoriadis, K.M. 206
Grote, W. 562
Guo, C. 1010

Haller, J. 275
Hanly, S.V. 786
Harris, J.M. 945
Hartmann, C.R.P. 690
Hegarty, C. 1063
Hegde, M.V. 552, 588
Hellal, O.A. 964
Helleseth, T. 480
Heymann, M. 136
Hirasawa, S. 333
Holloway, L.E. 146
Honig, M.L. 925
Hou, L. 349
Humes, C. 510

Ishida, Y. 458

Jacques, D.R. 196
Jean-Marie, A. 964
Jelenković, P.R. 746
Ji, C. 545, 994
Ji, H. 1093
Jiang, H. 255
Jordan, H.F. 568

Jordan, S. 255
Joshi, S. 275

Kabanza, F. 96
Kambhamettu, C. 658
Kandlur, D.D. 935
Kang, I. 245
Kasami, T. 700
Kaveh, M. 216
Kepler, J.F 796
Khosla, P.K. 678
Kieffer, J.C. 881
Kim, J.B. 925
Kløve, T. 714
Koç, Ç. 440
Kofman, D. 974
Kohno, R. 776
Konstantopoulos, I.K. 69
Koole, G. 736
Krishna, A. 935
Krogmeier, J.V. 235
Kuh, A. 1010
Kulkarni, S.R. 1033
Kumar, P.V. 480
Kumar, R. 166
Kumar, S.P. 945
Kumar, V. 116
Kurtas, E. 323

Ladner, R. 265
Langari, A. 401
Lazar, A.A. 746
Leduc, R.J. 519
Lehnert, J.S. 225, 805
Li, W.-C. W. 475
Lin, F. 136
Lin, S. 700
Lin, W. 313
Little, J. 492
Liu, Q. 466
Liu, R.-w. 1020
Liu, Z. 736
Lok, T.M. 805
Love, C.E. 568
Lu, S. 285
Lyon, D.E. 421

Ma, S. 994
Maier, R.S. 766
Malloy, D. 49
Mandayam, N.B. 1043
Marić, S. 911
Masry, E. 529
Mattei, M. 59
Maunder, A. 588

Mazumdar, R. 726
McEliece, R.J. 313, 366
McKellips, A. 608
McKeown, N. 984
Melsa, P.J.W. 431
Meyer, G.G.L. 447
Michel, A.N. 349, 535
Miller, M.I. 275
Min, P.S. 552, 588
Mittelholzer, T. 380
Modha, D.S. 529
Mohan, C.K. 690
Moody, J.O. 176
Moreno, O. 911

Narayan, A. 295
Nazmul Karim, M. 340
Nelson, B.J. 678
Nelson, L.B. 216
Nishijima, T. 333
Nogi, A. 458
Noneaker, D.L. 825

Ozbay, H. 853

Panwar, S.S. 562
Park, Y. 126
Pascale, M. 447
Patel, B.V. 955
Peery, T.E. 853
Perez, L.C. 357
Pimentel, C. 648
Poonen, B. 475
Posner, S.E. 1033
Prabhakar, B. 984
Proakis, J.G. 323
Pursley, M.B. 825

Raichle, M.E. 275
Rao, T.R.N. 483
Rasky, P.D. 796
Rasmussen, L.K. 21, 29
Ratsaby, J. 1002
Ricker, S.L. 156
Ridgely, D.B. 196
Rimoldi, B. 39
Riskin, E.A. 265
Roberts, J.W. 726
Rodemich, E.R. 366
Rohrs, C.E. 431
Rokhsaz, K. 79
Rosenberg, C. 724
Rudie, K. 156
Rump, C.M. 756

Saidi, H. 552
Salehi, M. 323
Salih, M. 1083
Sarkar, N. 156
Sawan, M.E. 79, 391
Schlegel, C. 29
Schwartz, M. 955
Seghers, J. 357
Sendonaris, A. 628
Setola, R. 59
Sexton, A. 421
Shanbhag, A.G. 480
Shankar, G. 303
Shi, G. 863
Shih, C.-C. 690
Shor, M.H. 411
Shue, S.-P. 79
Sidhu, I. 255
Simonian, A. 726
Skelton, R.E. 863
Srinivas, B.S. 265
Stark, W.E. 815
St-Denis, R. 96
Stidham, S. 756
Szpankowski, W. 891

Tang, J. 891
Tantaratana, S. 1073, 1083
Tarokh, V. 710
Théberge, F. 726
Tolli, D.M. 796
Tse, D.N. 786

Varaiya, P. 106
Varanasi, M.K. 11, 1053
Varvarigos, E. 578
Veeravalli, V. 628
Venkatesh, S.S. 1002
Verdú, S. 608, 1043
Vojcic, B. 1063

Wang, K. 535
Wang, T.-c. 391
Wei, V.K. 901
Wonham, W.M. 501, 519
Wulff, C.R. 690

Yamamoto, H. 700
Yang, E.-h. 466, 881, 901
Yang, K. 480
Yang, Y. 1028
Yaz, E.E. 835
Ye, H. 349, 535
Younce, R.C. 431
Young, P.M. 186
Ytrehus, Ø. 382

Zhang, L. 146
Zhang, Y. 618
Zhang, Z. 466, 901

UNIVERSITY OF ILLINOIS-URBANA
3 0112 033279768